ADVANCES IN ONLINE CONVEX OPTIMIZATION, GAMES, AND PROBLEMS WITH BANDIT FEEDBACK

A Dissertation
Presented to
The Academic Faculty

By

Adrian Rivera Cardoso

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Industrial and Systems Engineering

Georgia Institute of Technology

May 2020

ADVANCES IN ONLINE CONVEX OPTIMIZATION, GAMES, AND PROBLEMS WITH BANDIT FEEDBACK

Approved by:

Dr. Huan Xu, Advisor
School of Industrial and Systems Engineering/Alibaba Group
Georgia Institute of Technology

Dr. He Wang, Advisor
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Rachel Cummings
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Jacob Abernethy
School of Computer Science
Georgia Institute of Technology

Dr. Sebastian Pokutta
Institute of Mathematics
Technische Universität Berlin

Date Approved: December 13, 2020

No Regrets not a single one.

To mom and dad,

ACKNOWLEDGEMENTS

First of all I would like to thank my advisor, Huan Xu. You were an excellent advisor in every possible sense. Thank you for all the guidance on how to learn a new topic starting from zero, on how to find find interesting problems and how to solve them. Thank you for all the academic freedom, financial support, and your infinite patience. Thank you for teaching me how to be a researcher. I also want to thank my advisor, He Wang. Thank you for teaching me how to push harder on a problem until it finally breaks, for all the advise on how to write papers. I also want to thank you for your infinite patience and moral support. I could not have been advised by a better pair of advisors. I also want to thank the rest of the committee, Rachel Cummings, Jacob Abernethy and Sebastian Pokutta for reading through this thesis, and for all the feedback they have given me during my PhD studies. Thank you Rachel, for introducing me to the field of Differential Privacy and for teaching me how to do research in that field. I enjoyed working with you. Thank you Jake for listening and for bouncing ideas with me. I like the way you think about online learning and I am very proud of the paper we wrote together. Lastly, thank you Sebastian for all the discussions after class, for all the great questions you have asked, and for the access to your computing infrastructure. I would also like to thank Arkadi Nemirovski, George Lan, and Shabbir Ahmed for all the optimization courses you taught, for receiving me in your offices without appointments, and for all the helpful research discussions.

I would also like to thank all the people at Roadie for the opportunity to intern there. In particular I would like to thank Daniel Zink and Kevin Ryan for letting me be part of their team, for everything you taught me, and for all the patience.

I would like to give special thanks to my friends Jana Boerger, Alejandro Carderera, Julia Lane, Kala Garapati, Yuliia Lut, and Sebastian Salazar, thank you for all the support and care you have given me these last couple of months. I would also like to thank all my friends here at Georgia Tech, Alfredo Torrico, Matias Siebert, Seyma Gurkan, Felipe Lagos, Sait

Cakmak, Ian Herszterg, Catharina Hollauer, Hassan Mortagy, Reid Bishop, Mohammed El Tonbari, John Connely, Andrew Elhabr, Digvijay Boob, Ramon Auad, Adolfo Rocco, Asteroide Santana, George Kotsalis, Amanda Chu, Will Lassiter, Beste Basciftci, Idil Arsik, Reem Khir, Daniela Hurtado, Timur Tankayev, Wanrong Zhang, Isabella Sanders, Adam Behrendt, and Arden Baxter you have made these last years very enjoyable. I would also like to thank Kendall Whitehurst; I still remember very clearly your words of support.

Most importantly I would like to thank my brother and my parents. This thesis would not have been possible without you. I have no words to describe everything have given me and done for me throughout my life. All I can say is thank you for all the support and the unconditional love. I love you with all my heart, this thesis is for you.

TABLE OF CONTENTS

LIST OF FIGURES

SUMMARY

In this thesis we study sequential decision making through the lens of Online Learning. Online Learning is a very powerful and general framework for multi-period decision making. Due to its simple formulation and effectiveness it has become a tool of daily use in multibillion companies. Moreover, due to its beautiful theory and its tight connections with other fields, Online Learning has caught the attention of academics all over the world and driven first-class research.

In the first chapter of this thesis, joint work with Huan Xu, we study a problem called: Risk-Averse Convex Bandit. Risk-aversion makes reference to the fact that humans prefer consistent sequences of good rewards instead of highly variable sequences with slightly better rewards. The Risk-Averse Convex Bandit addresses the fact that, while human decision makers are risk-averse, most algorithms for Online Learning are not. In this thesis we provide the first efficient algorithms with strong theoretical guarantees for the Risk-Averse Convex Bandit problem.

In the second chapter, joint work with Rachel Cummings, we study the problem of preserving privacy in the setting of online submodular minimization. Submodular functions have multiple applications in machine learning and economics, which usually involve sensitive data from individuals. Using tools from Online Convex Optimization, we provide the first ϵ-differentially private algorithms for this problem which are almost as good as the non-private versions for this problem.

In the third chapter, joint work with Jacob Abernethy, He Wang, and Huan Xu, we study a dynamic version of two player zero-sum games. Zero-sum games are ubiquitous in economics, and central to understanding Linear Programming Duality, Convex and Robust Optimization, and Statistics. For many decades it was thought that one could solve this kind of games using sublinear regret algorithms for Online Convex Optimization. We show that while the previous is true when the game does not change with time, a naive application

of these algorithms can be fatal if the game changes and the players are trying to compete with the Nash Equilibrium of the sum of the games in hindsight.

In the fourth chapter, joint work with He Wang and Huan Xu, we revisit the decade old problem of Markov Decision Processes (MDPs) with Adversarial Rewards. MDPs provide a general mathematical framework for sequential decision making under uncertainty when there is a notion of 'state', moreover they are the backbone of all Reinforcement Learning. We provide an elegant algorithm for this problem using tools from Online Convex Optimization. The algorithm's performance is comparable with current state of the art. We also consider the problem under the large state-space regime, and provide the first algorithm with strong theoretical guarantees.

CHAPTER 1
INTRODUCTION

Online Learning is a very powerful and general framework for multi-period decision making. Due to its simple formulation and effectiveness it has become a tool of daily use in multibillion companies such as Google, Microsoft, Amazon, Alibaba, Netflix, Uber, etc. Additionally, due to its beautiful theory and its tight connections with Machine Learning, Optimization, Game Theory, Differential Privacy, Finance, and Operations Research, Online Learning has caught the attention of academics all over the world and driven first-class research.

The most general description of the Online Learning setting is the following. Consider a player that repeatedly has to make a decision at the beginning of each period without knowing how good it will be. Once the decision has been made, the player receives some feedback from the environment about how good the decision was. With this new information, and with all the feedback the player has received up to that point it must make a decision for the next time period, and so on. The goal of the player is to, of course, choose the best action it could have taken for that period. However, in this very general setting, without knowing anything about how good its action will be, it is impossible for the player to achieve this goal. Because of the previous, the player will measure how good its actions were according to how big its regret was at the end of the game.

1.1 A Brief History of Online Learning

To the best of my knowledge, the first paper in what is now considered Online Learning was that of William R. Thompson in 1933 [125]. Thompson studied the problem of how to adaptively sample from two distributions to determine which of the distributions was better than the other. In 1947, Wald published a book titled "Sequential Analysis" [130] where

he studied the studied the general problem of 'statistical inference where the number of samples is not fixed in advance', which was posed to him by economists Milton Friedman and Allen Wallis in 1943. Other important pieces of work that came out after were those of K. J. Arrow, D. Blackwell, and M. A. Girshick [13], and H. Robbins [111]. Up to this point all the research done in sequential analysis involved choosing from an action from a finite set, and the observed outcomes were random. In the next couple of decades researchers coined the term "Multi-Armed Bandit" to describe the problem where a decision maker must repeatedly choose an action (from a fixed set of alternatives), observe only the corresponding reward, and try to maximize its cumulative reward using only the information available at that point. The term Multi-Armed Bandit comes from the fact that people called slot machines in casinos "one-armed bandits", as they slowly take your money.

The first one to deviate from stochastic assumptions was James Hannan in 1957 [67]. He assumed that the outcome of his actions could be assigned arbitrarily by nature and devised a strategy for choosing actions such that the average 'inutility' of the actions was close to that as if he had known the state of nature in advance. In other words he devised the first strategy that ensured low regret in an adversarial setting.

In 1985, Lai and Robins in their seminal paper [90] presented the first asymptotically optimal strategy for the stochastic Multi-Armed Bandit.

It was not until 1988, when research under adversarial assumptions started being of interest again. Some of this works include those of Littlestone, Vovk, Foster and Cover [42, 55, 92, 129]. Of particular interest is the work of Littlestone and Warmuth (1994) where they considered the "experts problem", a problem very similar to the Multi-Armed Bandit with the difference that the rewards are adversarial instead of stochastic and the decision maker gets to observe the rewards of all the arms, even of the ones it did not select. From then on, a great amount of research in the area was done. In fact, so much that I will not even try to summarize it but instead I will refer you to recent surveys by Elad Hazan, Nicolò Cesa-Bianchi, Gabor Lugosi and Sébastien Bubeck [32, 36, 72].

In 2003 Martin Zinkevich published [139] where he formally defined the setup of Online Convex Optimization. The general setup of Zinkevich generalized and unified a great amount of work published before him. The setup was that of Online Learning described previously where the decision set was a convex and bounded set and the loss functions were convex and fully revealed to the player after the decision had been made. He also presented an algorithm which we know as online gradient descent that is optimal with respect to the number of time periods. Three years later Abraham Flaxman, Adam Kalai and Brendan McMahan [53] studied Online Bandit Optimization, essentially the same problem as Online Convex Optimization with the difference that the feedback is much weaker. Instead of observing the whole convex loss function they would only observe the value of the function evaluated at a given point. This kind of feedback, called bandit feedback, is more realistic in many applications where you can only observe the consequences of your actions and not what would have happened if you had taken another action. However, the algorithm they presented "Online Gradient Descent without a Gradient", was not optimal with respect to the number of rounds. It was not until 2016 that two groups of people, the first one composed by Sebastien Bubeck, Yin Tat Lee and Ronen Eldan and the second one by Elad Hazan and Yuanzhi Li designed polynomial time algorithms for Online Bandit Optimization that were optimal with respect to the number of rounds [33, 70].

1.2 Overview of Results

As practitioners have adopted the tools from Online Learning in different areas such as finance, clinical trials, online advertising, and other Internet applications a need has arisen to fine-tune the tools and generalize them to account for factors such as: the privacy of the users, the risk level of the decision makers, the changing dynamics of the decision environment, and other additional constraints dependent on the application. In this thesis we aim to do exactly such a thing.

In Chapter 2 motivated by applications in clinical trials and finance, we study the prob-

lem of Online Bandit Optimization where the decision maker is risk-averse. We provide two algorithms to solve this problem. The first one is a descent-type algorithm which is easy to implement. The second algorithm, which combines the ellipsoid method and a center point device, achieves (almost) optimal regret bounds with respect to the number of rounds. We provide theoretical guarantees for these algorithms as well as some experimental results. To the best of our knowledge this is the first attempt to address risk-aversion in the online convex bandit problem.

In Chapter 3 motivated by online recommenders systems, we develop the first algorithms for online submodular minimization that preserve differential privacy under full information feedback and bandit feedback and provide theoretical guarantees.

In Chapter 4 we study zero-sum games where the payoffs may change arbitrarily, this is a generalization of Online Convex Optimization which explicitly brings a second player into the picture. At each iteration a pair of actions need to be chosen without knowledge of the future (convex-concave) payoff functions. The objective is to minimize the gap between the cumulative payoffs and the Nash Equilibrium value of the aggregate payoff function, which we measure using a metric called Nash Equilibrium Regret. The problem can be interpreted as trying to compete with the Nash equilibrium for the aggregate of a sequence of two-player zero-sum games without having any previous knowledge of the games. We also study the problem under the more challenging setting of bandit feedback.

In Chapter 5 we study Markov Decision Processes (MDPs) with Adversarial Rewards. MDPs provide a general mathematical framework for sequential decision making under uncertainty when there is a notion of 'state', moreover they are the backbone of all Reinforcement Learning. We provide an elegant algorithm for this problem using tools from Online Convex Optimization. The algorithm's performance is comparable with current state of the art. We also consider the problem under the large state-space regime, and provide the first algorithm with strong theoretical guarantees.

CHAPTER 2

RISK-AVERSE CONVEX BANDIT

2.1 Introduction

In this chapter we study the problem of Online Risk-Averse Optimization which generalizes Online Convex Optimization (OCO) and Online Bandit Optimization (OBO). The standard goal of OCO and OBO is to develop algorithms such that the standard average regret

$$\frac{1}{T}\sum_{t=1}^{T} f_t(x_t) - \frac{1}{T}\min_{x\in X}\sum_{t=1}^{T} f_t(x)$$

vanishes as quickly as possible. In other words, we want our average loss to be as close as possible to the best loss if we had known all the functions in advance and committed to one action. Here the sequence of convex functions $\{f_t\}_{t=1}^{T}$ may be chosen by an adversary and the regret minimizing algorithm chooses action x_{t+1}, in some bounded convex set X, by using only the information available at time t. This means that in the OCO setting the algorithm may use $\{x_1, ..., x_t\}$ and $\{f_1(\cdot), ...f_t(\cdot)\}$, and in the OBO setting it may only use $\{x_1, ..., x_t\}$ and $\{f_1(x_1), ...f_t(x_t)\}$. While the set up of OCO and OBO is very powerful because it allows for the loss functions to be chosen adversarially, in some applications such as medicine and finance this may not be enough.

Let us consider an example in clinical trials. Suppose there are T patients with some rare disease and we have at our disposal a new drug that has the potential to cure the disease if we prescribe the right dose. Since we do not know what the right dose is, we must learn it as we treat each patient. In other words, we will choose a dose, observe the reaction of a patient and chose a new dose for the next patient. The previous problem can of course be be abstracted as an OBO problem, where each function $f_t(\cdot)$ encodes how patient t will react to the dose we prescribe x_t. Here, the assumption that f_t is chosen adversarially

may not be very realistic and perhaps it makes more sense to assume that f_t is drawn randomly from some family of functions. An algorithm that guarantees that the standard average regret vanishes can be seen as an algorithm that is choosing the optimal dose for the average patient, something that is non-trivial to do. Unfortunately, such guarantee completely ignores what may happen to patients that do not look like the average patient. It could be that the optimal dose for the average patient has really negative effects on 5% of the patients. In this case, a dose that is slightly less effective on the average patient but does not harm the unlucky 5% may be more desirable. Thus, the goal of this chapter is to provide algorithms for OCO and OBO that *explicitly incorporate risk*. By "risk" we mean the possibility of really negative outcomes, as it is used in the Economics and Operations Research communities.

Another area where an explicit consideration of risk must be taken into account is finance. For example, in [49] the authors show that in the online portfolio problem, risk neutral guarantees such as performing as well as the best constant rebalanced portfolio (i.e. minimizing standard average regret) may not perform well in practice. They show through experiments on the S&P500 that the simple strategy that maintains uniform weights on all the stocks outperforms that which seeks to perform as well as the best stock (regardless of its theoretical guarantees). To explicitly incorporate risk into the setting of OCO and OBO we will use a coherent risk measure called Conditional Value at Risk ($CVaR$) [112], sometimes also called Expected Shortfall, which is widely used in the financial industry. After the financial crisis of 2008, the Basel Committee on Banking Supervision created the Third Basel Accord (Basel III), a set of regulatory measures to strengthen the regulation, supervision and risk management of the banking sector [54]. In this accord one of the main points was to migrate from quantitative risk measures such as Value at Risk to Conditional Value at Risk since it better captures tail risk.

It should be clear from the previous examples that generally speaking, human decision makers are risk-averse. They prefer consistent sequences of rewards instead of highly vari-

able sequences with slightly better rewards. Because of the previous, we want to develop algorithms that explicitly incorporate risk which have strong theoretical guarantees.

2.1.1 Main Results

Our main contributions are the following. First, we develop and analyze two algorithms for the online convex bandit problem that explicitly incorporate the risk aversion of the decision maker (as measured by the $CVaR$). Second, we extend our results to the case where the decision maker uses more general risk measures to measure risk by using the Kusuoka representation theorem.

2.2 Related Work

Risk aversion has received very little attention in the online learning setting. The few existing work all focuses on the case where *the number of actions is finite*: For the stochastic multi-armed bandit problem, [114] provide algorithms that ensure the mean-variance of the sequence of rewards generated by the algorithm is not too far from the mean-variance of the rewards generated by the best arm. In [127] the same problem is studied and the authors provide tighter upper and lower bounds. In [97] the author considers a different risk measure, the cumulant generative function, and provide similar guarantees for a slightly modified definition of regret. In [60] the authors consider the $CVaR$ as measure of risk aversion and provide algorithms that achieve sublinear regret. The notion of regret they use is different from the one we will use as they do not look at the risk of the sequence of rewards obtained by the algorithms, but instead they seek to perform as well as the arm that minimizes $CVaR$ (i.e., "pseudo regret" as we called). However recall that they only consider a finite number of arms, whereas we consider an uncountably infinite number of arms. In [135] the authors study the related problem of best arm identification where the goal is to identify the arm with the best risk measure. They consider Value at Risk, $CVaR$, and Mean-Variance as risk measures. In [49] the authors consider risk aversion in the ex-

perts problem. This setting is similar to the multi-armed bandit problem with the difference that the rewards are assigned adversarially, and at each time step all the rewards are visible to the player. In particular they seek to build algorithms such that the mean variance (or Sharpe ratio) of the sequence of rewards generated by the algorithm are as close as possible to that of the best expert. They show negative results for this problem however they provide algorithms that perform well for "localized" versions of the risk measures they consider.

To the best of our knowledge, all existing work that explicitly incorporates risk aversion under the assumptions of stochastic rewards and bandit feedback is restricted to the multi-armed bandit model. This work is the first to consider an infinite number of arms and incorporate risk aversion under bandit feedback. In [49], where risk aversion in the experts problem is studied, one can think of instead of choosing an expert at every round one chooses a probability distribution over the experts. While the set of probability distributions over the experts is a convex set, this is a very specialized case (linear functional and simplex feasible set). Moreover, the authors assume full information feedback and adversarial rewards, which are very different from our setup.

2.3 Preliminaries

This section is devoted to preliminaries. In particular we review relevant concepts and technical results essential to develop the proposed algorithms.

2.3.1 Notation

Let $||\cdot||$ be the l_2 norm unless otherwise stated. By default all vectors are column vectors, a vector with entries $x_1, ..., x_n$ is written as $x = [x_1; ...; x_n] = [x_1, ..., x_n]^\top$ where $\top$ denotes the transpose. For a random variable X, $X \sim P$ means that X is distributed according to distribution P. We let $\nabla g(x)$ be any element in the subdifferential of g at x. Whenever we write $\nabla f(x, \xi)$ we mean $\nabla_x f(x, \xi)$. Throughout the chapter we will use O notation to hide constant factors, when appropriate we will make use of a universal constant κ to represent

the constant factor (i.e. $O(T) = \kappa T$), this constant κ may change from line to line. We use $\tilde{O}$ notation to hide constant factors and poly-logarithmic factors of T, $\frac{1}{\alpha}$ and d.

2.3.2 Convexity and Lipschitz Continuity

Let $X \subseteq \mathbb{R}^d$ be a convex set, that is, for any $x, y \in X$ and any $\lambda \in [0, 1]$, $\lambda x + (1-\lambda)y \in X$. We say $f : X \to \mathbb{R}$ is a convex function if for any $\lambda \in [0, 1]$ and for any $x, y \in X$

$$\lambda f(x) + (1 - \lambda)f(y) \geq f(\lambda x + (1 - \lambda)y).$$

An equivalent definition of convexity is the following [104]. f is convex if and only if

$$f(x) \geq f(y) + \nabla f(y)^\top (x - y) \quad \forall x, y \in X.$$

Here $\nabla f(y)$ denotes any element in the subdifferential of f at y.

We say $f : X \to \mathbb{R}$ is strongly convex with parameter $\beta > 0$ if and only if

$$f(x) \geq f(y) + \nabla f(y)^\top (x - y) + \frac{\beta}{2}||x - y||^2 \quad \forall x, y \in X.$$

We say f is G-Lipschitz continuous with respect to a norm $|| \cdot ||$ if for every $x, y \in X$, $|f(x) - f(y)| \leq G||x - y||$.

Lemma 1. *[117] [Ch. 2]Let $f : X \to \mathbb{R}$ be a convex function. Then, f is G-Lipschitz over X with respect to a norm $|| \cdot ||$ if and only if for all $x \in X$ and for all $\nabla f(x) \in \partial f(x)$ we have that $||\nabla f(x)||_* \leq G$, where $|| \cdot ||_*$ denotes the dual norm.*

Throughout this section, whenever we say f is G-Lipschitz we mean f is G-Lipschitz with respect to $|| \cdot ||_2$ unless otherwise stated.

2.3.3 From OCO to to Bandit Feedback

We present a result from that allows us to transform regret bounds from OCO into expected regret bounds for Online Bandit Optimization.

Lemma 2. *[72][Ch. 6] Let u be a fixed point in X. Let $f_1, ..., f_T : X \to \mathbb{R}$ be a sequence of differentiable functions. Let $\mathcal{A}$ be a first order algorithm that ensures Regret$_T(\mathcal{A}) \leq B_{\mathcal{A}}(\nabla f_1(x_1), ..., \nabla f_T(x_T))$ in the full information setting. Define $\{x_t\}$ as: $x_1 \leftarrow \mathcal{A}(\emptyset)$, $x_t \leftarrow \mathcal{A}(g_1, ..., g_{t-1})$ where each g_t satisfies:*

$$\mathbb{E}[g_t | x_1, f_1, ..., x_t, f_t] = \nabla f_t(x_t)$$

Then, for every $u \in X$:

$$\mathbb{E}[\sum_{t=1}^{T} f_t(x_t)] - \sum_{t=1}^{T} f_t(u) \leq \mathbb{E}[B_{\mathcal{A}}(g_1, ...g_T)]$$

Moreover, Online Gradient Descent is a first order Algorithm [72][Ch. 6].

2.3.4 One-Point Gradient Estimation

Consider function $f : \mathbb{R}^d \to \mathbb{R}$ which is G-Lipschitz continuous. Define its smoothened version

$$\hat{f}^{\delta}(x) := \mathbb{E}_{v \sim \mathbb{B}}[f(x + \delta v)]$$

where $\mathbb{B}$ is the uniform distribution over the unit ball of appropriate dimension. From now on we omit superscript δ and write $\hat{f}(x)$. Define random quantity

$$g = \frac{d}{\delta} f(x + \delta u) u \tag{2.1}$$

with $u \sim \mathbb{S}$ where $\mathbb{S}$ is the uniform distribution over the unit sphere. We have the following

Lemma 3. *[72][Ch.2] $\hat{f}$ satisfies the following:*

1. *If f is α-strongly convex then so is $\hat{f}$*
2. $|f(x) - \hat{f}(x)| \leq \delta G$
3. $\mathbb{E}[g] = \nabla \hat{f}(x)$

That is, the smoothened version of f is convex as well, it is not too far from f, and by sampling from the unit sphere we can obtain an unbiased estimate of its gradient.

2.3.5 Conditional Value at Risk

In [112] the authors define the α-Value at Risk of random variable X as

$$VaR_\alpha[X] := \inf\{t : P(X \leq t) \geq 1 - \alpha\}.$$

Using the above definition they define Conditional Value at Risk ($CVaR$, sometimes also called Expected Shortfall) as

$$C_\alpha[X] := CVaR_\alpha[X] := \frac{1}{\alpha} \int_{1-\alpha}^{\alpha} VaR_{1-\tau}[X] d\tau. \tag{2.2}$$

Moreover, when the random variable has c.d.f $H(x)$ continuous at $x = VaR_\alpha[X]$ it holds that

$$C_\alpha[X] = \mathbb{E}[X | X \geq VaR_\alpha[X]]. \tag{2.3}$$

We make use of the following notation. Let $\{a_t\}_{t=1}^T$ be an arbitrary sequence of real numbers, we let $C_\alpha[\{a_t\}_{t=1}^T]$ be the Conditional Value at Risk of the discrete random variable that takes each value a_t with probability $1/T$ for all $t = 1, ..., T$.

Below we state some well known results that will be used later. The proofs for the next two lemmas can be found in [119].

Lemma 4.

$$CVaR_\alpha[X] = \min_{z\in\mathbb{R}} z + \frac{1}{\alpha}\mathbb{E}[X - z]_+, \tag{2.4}$$

where $[a]_+ := \max\{a, 0\}$. *In fact, if* $0 \leq X \leq 1$ *with probability 1, the condition* $z \in \mathbb{R}$ *can be replaced with* $z \in [0, 1]$.

Lemma 5. *Let* ξ *be a random variable supported in* Ξ *with distribution* P, *let* $X \subset \mathbb{R}$ *be a convex and compact and let* $f : X \times \Xi \to \mathbb{R}$ *be convex in* x *for every* ξ. *Define* $F = f(x, \xi)$. *Then*

$$C_\alpha[F](x) := CVaR_\alpha[F](x) = \min_z z + \frac{1}{\alpha}\mathbb{E}_\xi[f(x, \xi) - z]_+$$

and $C_\alpha[F](x)$ *is a convex function of* x. *In fact, if* $f(\cdot, \xi)$ *is* β*-strongly convex for every* $\xi \in \Xi$, *then so is* $C_\alpha[F](x)$.

2.4 Problem Setup

In this section we formally define the setup of our problem. Let ξ be a random variable supported in Ξ with unknown distribution P. Let $X \subset \mathbb{R}^d$ be a convex and compact set with diameter D_X that contains the origin. Let $f : X \times \Xi \to \mathbb{R}$ be a convex function in the first argument for every $\xi \in \Xi$. Let f satisfy $||\nabla f(x, \xi)|| \leq G$ for every $x \in X$ and every $\xi \in \Xi$. We define random function $F(x) = f(x, \xi)$ in the sense that for every $x \in X$, $F(x)$ is a random variable. We also assume that for every $x \in X$, $0 \leq F(x) \leq 1$ with probability 1.

A risk-averse player will make decisions in a *stochastic environment* for T time steps. In every time step $t = 1, ..., T$ the player chooses action $\tilde{x}_t \in X$, and nature obtains sample ξ_t from P. Then, the player incurs and observes only the loss incurred by its action $f(\tilde{x}_t, \xi_t)$. If the player were risk neutral then a reasonable goal would be to design an algorithm that

obtains (in expectation) vanishing standard average regret, that is

$$\mathbb{E}[\frac{1}{T}\sum_{t=1}^{T} f(\tilde{x}_t, \xi_t) - \frac{1}{T}\min_{x\in X}\sum_{t=1}^{T} f(x, \xi_t)] = o(1).$$

Where the expectation is taken with respect to the random draw of functions and the internal randomization of the algorithm. Such is the standard goal of OCO and OBO, and as mentioned in the introduction, there already exist polynomial time algorithms that achieve the optimal lower bound of $\Omega(1/\sqrt{T})$ (up to logarithmic factors) even when the functions f are chosen by an adversary instead of from some distribution.

In our setting, since the player is risk averse, the notion of average regret is not appropriate. In this section we assume that the player uses the Conditional Value at Risk $C_\alpha[\cdot] = CVaR_\alpha[\cdot]$ for some $\alpha \in (0, 1]$ to measure risk (when $\alpha = 1$, $C_\alpha[\cdot] = \mathbb{E}[\cdot]$ i.e. the player becomes risk neutral). With this in mind, the following two quantities become interesting, namely pseudo-$CVaR$-regret defined as

$$\bar{\mathcal{R}}_T := \frac{1}{T}\sum_{t=1}^{T} C_\alpha[F](\tilde{x}_t) - \frac{1}{T}\min_{x\in X}\sum_{t=1}^{T} C_\alpha[F](x) \tag{2.5}$$

and $CVaR$-regret defined as

$$\mathcal{R}_T := C_\alpha[\{f_t(\tilde{x}_t)\}_{t=1}^T] - \min_{x\in X} C_\alpha[\{f_t(x)\}_{t=1}^T],$$

where we make more explicit what we mean by $C_\alpha[\{f_t(x_t)\}_{t=1}^T]$ in the next paragraph. In this setup, a risk averse player may be concerned with two types of risk, the risk of the individual losses it incurs and the overall risk of playing the game. The player that is concerned about the risk of the individual losses, should be pleased with an algorithm that obtains vanishing $\bar{\mathcal{R}}_T$, this would ensure that the average risk of the losses it incurs is not too far from that of the best point in the set.

On the other hand, the player that is concerned about the overall risk of playing the

game may desire a different guarantee. Notice that the sequence of losses that the player incurs $\{f_t(\tilde{x}_t)\}_{t=1}^T$ defines an empirical distribution where every realization $f_t(\tilde{x}_t)$ occurs with probability $\frac{1}{T}$ and as such we can compute its risk $C_\alpha[\{f_t(\tilde{x}_t)\}_{t=1}^T]$. It is then natural for the player to desire a sequence of losses that has risk as close as possible to the minimum risk sequence of losses (where the sequence is generated by playing only one action). The quantity $\mathcal{R}_T$ makes the previous statement precise.

A reader familiar with the OBO literature may notice that (2.5) already looks like a quantity for which running Online Gradient Descent without a Gradient may yield vanishing regret. Unfortunately, at every step all we observe is $f_t(\tilde{x}_t)$ and not $C_\alpha[F](\tilde{x}_t)$. To obtain a reasonable (not too noisy) evaluation of $C_\alpha[F](\cdot)$ the same x must be played for several rounds. It is possible to design algorithms that follow this idea, however, since we were able to develop better algorithms for the same problem we do not further discuss the details of this somewhat naive approach.

2.5 A Finite-Time Concentration Result for the $CVaR$

Before we present the algorithms we must derive a finite-time concentration result for the $CVaR$. This result will be heavily used to prove sublinear regret bounds for both algorithms. In [119] the authors present an asymptotic result. Unfortunately, since our goal is to achieve finite-time bounds we could not use it and had to prove our own result. To the best of our knowledge this is the first finite time concentration result for the $CVaR$.

Theorem 1. *Suppose* $0 \leq f(x,\xi) \leq 1$ *for every* $x \in X$ *and every* $\xi \in \Xi$ *. For any* $x \in X$*, let the N-sample estimate of* $CVaR_\alpha[F](x)$ *be* $\widehat{CVaR_\alpha[F]}(x) := \min_{z \in Z} z + \frac{1}{\alpha N}\sum_{n=1}^N [f(x,\xi_n) - z]_+$*. Where* $Z := [0,1]$*. It holds that with probability at least* $1-\delta$*,*

$$|CVaR_\alpha[F](x) - \widehat{CVaR_\alpha[F]}(x)| \leq O(\sqrt{\frac{\ln(N/\delta)}{\alpha^2 N}}).$$

While the previous result holds with high probability it is also possible to derive from

it a result that holds in expectation.

To prove such a result we had to use a finite time concentration result for Lipschitz functions from [118] applied to the sequence of functions $\{z + \frac{1}{\alpha}[f(x,\xi_t) - z]_+\}_{t=1}^T$. After this, some extra work had to be done transform this guarantee into one that holds for the $CVAR$. A formal proof of the theorem can be found next.

Proof of Theorem 1. For any fixed $x \in X$, we define $\phi(z) := z + \frac{1}{\alpha}E_{\xi\sim P}[f(x,\xi) - z]_+$ and $\widehat{\phi}(z) = \frac{1}{N}\sum_{n=1}^N z + \frac{1}{\alpha}[f(x,\xi_n) - z]_+$. By Lemma 64 we know that with probability at least $1-\delta$ for all $z \in [0,1]$

$$|\phi(z) - \widehat{\phi}(z)| \leq O(\sqrt{\frac{LR\ln(N/\delta)}{N}})$$

and it is easy to see that L, R are both $O(\frac{1}{\alpha})$.

It remains to show that $A := \{X_A = \sup_z |\phi(z) - \widehat{\phi}(z)| \leq \epsilon\}$ implies $B := \{X_B = |CVaR_\alpha[F](x) - \widehat{CVaR_\alpha[F]}(x)| \leq \epsilon\}$. Indeed, we have that for any $z \in Z$

$$\phi(z) - \epsilon \leq \widehat{\phi}(z)$$

Therefore, if $\bar{z} = \arg\min_{z\in Z} \widehat{\phi}(z)$ we have:

$$CVaR_\alpha[F](x) - \epsilon \leq \phi(\bar{z}) - \epsilon \leq \widehat{\phi}(\bar{z}) = \widehat{CVaR_\alpha}[F](x)$$

The other side of the inequality follows by applying the same type of argument to $\widehat{\phi}(z) \leq \phi(z) + \epsilon$. □

Remark 1. *We make one last remark about the proof above. We showed that $A \implies B$ therefore $P(B') \leq P(A')$. Since for a nonnegative random variable X we can write $\mathbb{E}[X] = \int P(X > \epsilon)d\epsilon$ we can conclude that $\mathbb{E}[X_B] \leq \mathbb{E}[X_A]$, or which is the same, $\mathbb{E}[|CVaR_\alpha[F](x) - \widehat{CVaR_\alpha[F]}(x)|] \leq \mathbb{E}[\sup_z |\phi(z) - \widehat{\phi}(z)|]$.*

2.6 Algorithm 1

In this section we provide an algorithm that obtains vanishing regret while playing an action only once. The key to the algorithm is to look at functions $\mathcal{L}_t(x, z) := z + \frac{1}{\alpha}[f(x, \xi_t) - z]_+$ which by Lemma 5 are closely related to $C_\alpha[F](x)$. Although with one sample we can not evaluate (accurately enough) $C_\alpha[F](\cdot)$, we can evaluate $\mathcal{L}_t$. This observation is important because it will allow us to build one-point gradient estimators of the smoothened function $\hat{\mathcal{L}}_t$ as it is done in [53]. These one-point gradient estimators will allow us to perform a descent step. This idea allows us to obtain sublinear pseudo-regret. The rest of the analysis consists of using the bound on the pseudo-regret to bound the regret.

Algorithm 1

Input: $X \subset \mathbb{R}^d$, $x_1 \in X$, $z_1 \in Z := [0, 1]$ step size η, δ
for $t = 1, ..., T$ **do**
  Sample $u \sim \mathbb{S}^{d+1}$
  Let $u^1 = [u_1; ...; u_d]$ and $u^2 = u_{d+1}$
  Play $\tilde{x}_t := x_t + \delta u^1$, incur and observe loss $f_t(\tilde{x}_t)$
  Let $\tilde{z}_t = z_t + \delta u^2$
  Let $g_t^1 := \frac{(d+1)}{\delta}(\tilde{z}_t + \alpha^{-1}[f_t(\tilde{x}_t) - \tilde{z}_t]_+)u^1$
  Let $g_t^2 := \frac{(d+1)}{\delta}(\tilde{z}_t + \alpha^{-1}[f_t(\tilde{x}_t) - \tilde{z}_t]_+)u^2$
  Update $x_{t+1} \leftarrow \Pi_{X_\delta}(x_t - \eta g_t^1)$
  Update $z_{t+1} \leftarrow \Pi_{Z_\delta}(z_t - \eta g_t^2)$
end for

Here $\mathbb{S}^d$ denotes the uniform distribution over the d-dimensional unit sphere, $X_\delta := \{x : \frac{1}{1-\delta}x \in X\}$ and $\Pi_X[\cdot]$ denotes the $||\cdot||_2$ projection onto convex set X.

We have the following two main results.

Theorem 2. *Using $\eta = \frac{\alpha D_{\mathcal{L}}}{(d+1)T^{3/4}}$ and $\delta = \frac{1}{T^{1/4}}$ Algorithm 1 guarantees:*

$$\mathbb{E}[\bar{\mathcal{R}}_T] \leq O(\frac{d}{\alpha T^{1/4}}).$$

Where the expectation is taken over the random draw of functions and the internal randomization of the algorithm. $D_{\mathcal{L}}$ is specified in the appendix.

Theorem 3. *Let $f(x, \xi)$ be strongly convex with parameter $\beta > 0$. Algorithm 1 guarantees*

$$\mathbb{E}[\mathcal{R}_T] \leq \tilde{O}(\frac{d^{1/2}}{\alpha^{3/2}\beta^{1/2}T^{1/8}}).$$

Where the expectation is taken over the random draw of functions and the internal randomization of the algorithm.

The proofs of these theorems can be found in the following subsection.

2.6.1 Analysis of Algorithm 1

Lemma 6. *The function $\mathcal{L}_t(x, z) := z + \frac{1}{\alpha}[f_t(x) - z]_+$ is jointly convex, $G_{\mathcal{L}}$-Lipschitz continuous with $G_{\mathcal{L}} = \alpha^{-1}(G + 1) + 1$, and the diameter of the set where it is defined $D_{\mathcal{L}} \leq D_X + 1$.*

Proof. We first prove convexity. The function $f_t(x) - z$ is jointly convex since both $f_t(x)$ and $-z$ are, and addition preserves convexity. Point-wise supremum over convex functions preserves convexity and since any constant function is convex we have that $[f_t(x) - z]_+$ is convex. Again, using the fact that addition preserves convexity we get the desired claim.

To prove the second part of the claim we notice:

$$\nabla_x \mathcal{L}_t(x, z) = \begin{cases} \frac{1}{\alpha}\nabla f_t(x) & \text{if } f_t(x) - z > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\nabla_z \mathcal{L}_t(x, z) = \begin{cases} 1 - \frac{1}{\alpha} & \text{if } f_t(x) - z > 0 \\ 1 & \text{otherwise} \end{cases}$$

Let $\nabla \mathcal{L}_t := [\nabla_x \mathcal{L}_t; \nabla_y \mathcal{L}_t]$ and recall that a function f is G-Lipschitz continuous if and only if $||\nabla f|| \leq G$. We have that We have that

$$||\mathcal{L}_t|| \leq \max\{|||[\bar{0}; 1]|||, |||[\alpha^{-1}\nabla f; 1 + \alpha^{-1}]|||\}$$

$$\leq \alpha^{-1}(G+1)+1 =: G_{\mathcal{L}}$$

Where the last inequality follows by simple algebra.

The fact that $D_{\mathcal{L}} \leq D_X + 1$ follows from the definition of the diameter of a set. □

The key to prove Theorem 2 is to realize that Algorithm 1 is performing Online Gradient Descent using an estimate of the gradient of the smoothened function $\hat{\mathcal{L}}_t$ as in [53].

Next we prove a lemma assuming that for every $t = 1, ..., T$ $\nabla\mathcal{L}_t := \nabla\mathcal{L}_t(x_t, z_t)$ is revealed and we update according to

$$[x_{t+1}, z_{t+1}]^\top \leftarrow \Pi_{X\times Z}([x_t, z_t]^\top - \eta\nabla\mathcal{L}_t) \tag{2.6}$$

That is, we perform Zinkevich's Online gradient Descent (OGD) on functions $\mathcal{L}_t$ [139]. Due to Lemma 2 we will be able to use this guarantee when we have bandit feedback.

Lemma 7. *Applying OGD on sequence of functions $\{\mathcal{L}_t\}_{t=1}^T$ guarantees: for every $w = (x, z) \in \mathcal{W} := X \times Z$.*

$$\sum_{t=1}^{T}\mathcal{L}_t(w_t) - \sum_{t=1}^{T}\mathcal{L}_t(w) \leq \frac{D_{\mathcal{L}}}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{T}||\nabla\mathcal{L}_t||^2.$$

Proof. We follow Zinkevich's proof. By properties of projections we have:

$$\begin{aligned}||w_{t+1} - w||^2 &\leq ||w_t - \eta\nabla\mathcal{L}_t - w||^2 \\ &= ||w_t - w||^2 + \eta^2||\nabla\mathcal{L}_t||^2 - 2\eta\nabla\mathcal{L}_t^\top(w_t - w)\end{aligned}$$

Therefore:

$$2\eta\nabla\mathcal{L}_t^\top(w_t - w) \leq \frac{||w_t - w||^2 - ||w_{t+1} - w||^2}{\eta} + \eta||\nabla\mathcal{L}_t||^2$$

Using convexity and summing up the inequalities above for every t we have:

$$2(\sum_{t=1}^{T}\mathcal{L}_t(w_t)-\sum_{t=1}^{T}\mathcal{L}_t(w)) \leq \sum_{t=1}^{T} 2\eta\nabla\mathcal{L}_t^\top(w_t-w) \tag{2.7}$$

$$\leq \sum_{t=1}^{T}\frac{||w_t-w||^2-||w_{t+1}-w||^2}{\eta}+\eta\sum_{t=1}^{T}||\nabla\mathcal{L}_t||^2 \tag{2.8}$$

$$\leq \frac{D_{\mathcal{L}}}{\eta}+\eta\sum_{t=1}^{T}||\nabla\mathcal{L}_t||^2$$

Which yields the desired result. □

Lemma 8. *Let $\tilde{y}_t=(\tilde{x}_t,\tilde{z}_t)$ and $y^*=(x^*,z^*):=$ argmin$_{x,z\in X\times Z}\sum_{t=1}^{T}\mathbb{E}_\xi[\mathcal{L}_t(x,z)]$, Algorithm 1 guarantees:*

$$\sum_{t=1}^{T}\mathbb{E}_{int}[\mathcal{L}_t(\tilde{y}_t)]-\sum_{t=1}^{T}\mathcal{L}_t(y^*)=O(\frac{dD_XGT^{3/4}}{\alpha})$$

Proof. Define $y_\delta^*=\Pi_{X_\delta}[y^*]$. By Lemma 67 in the Appendix, it holds that $||y_\delta^*-y^*||\leq\delta D_{\mathcal{L}}$. Using a similar argument as in [53] we have:

$$\begin{aligned}
&\mathbb{E}_{int}[\sum_{t=1}^{T}\mathcal{L}_t(\tilde{y}_t)-\sum_{t=1}^{T}\mathcal{L}_t(y^*)]\\
&\leq \mathbb{E}_{int}[\sum_{t=1}^{T}\mathcal{L}_t(y_t)-\sum_{t=1}^{T}\mathcal{L}_t(y^*)]+\delta G_{\mathcal{L}}T \quad \text{by Lemma 3 and } ||y_t-\tilde{y}_t||\leq\delta\\
&\leq \mathbb{E}_{int}[\sum_{t=1}^{T}\mathcal{L}_t(y_t)-\sum_{t=1}^{T}\mathcal{L}_t(y_\delta^*)]+\delta G_{\mathcal{L}}T+\delta G_{\mathcal{L}}D_{\mathcal{L}}T\\
&\leq \mathbb{E}_{int}[\sum_{t=1}^{T}\hat{\mathcal{L}}_t(y_t)-\sum_{t=1}^{T}\hat{\mathcal{L}}_t(y_\delta^*)]+3\delta G_{\mathcal{L}}T+\delta G_{\mathcal{L}}D_{\mathcal{L}}T \quad \text{by Lemma 3}\\
&\leq \frac{\eta}{2}\sum_{t=1}^{T}\mathbb{E}_{int}[||g_t||^2]+\frac{D_{\mathcal{L}}^2}{2\eta}+3\delta G_{\mathcal{L}}T+\delta D_{\mathcal{L}}G_{\mathcal{L}}T \quad \text{by Lemma 2}\\
&\leq \frac{\eta}{2}\frac{(d+1)^2}{\delta^2}\sum_{t=1}^{T}|\tilde{z}_t+\frac{1}{\alpha}[f_t(\tilde{x}_t)-\tilde{z}_t]|^2+\frac{D_{\mathcal{L}}^2}{2\eta}+3\delta G_{\mathcal{L}}T+\delta D_{\mathcal{L}}G_{\mathcal{L}}T\\
&\leq \frac{\eta}{2}\frac{(d+1)^2}{\delta^2\alpha^2}T+\frac{D_{\mathcal{L}}^2}{2\eta}+3\delta G_{\mathcal{L}}T+\delta D_{\mathcal{L}}G_{\mathcal{L}}T
\end{aligned}$$

$$= O(\frac{dD_X G T^{3/4}}{\alpha})$$

Where we chose $\eta = O(\frac{D_X \alpha}{dT^{3/4}})$ and $\delta = O(\frac{1}{T^{1/4}})$. □

We are now ready to give a proof of Theorem 2.

Proof of Theorem 2. Notice that for all t, every $x \in X$ and every $z \in Z$, we have:

$$\mathbb{E}_{\xi \sim P}[\mathcal{L}_t(x, z)] = z + \frac{1}{\alpha}\mathbb{E}_{\xi \sim P}[f(x, \xi) - z]_+ \geq CVaR_\alpha[F](x).$$

The result then follows by taking $\mathbb{E}_{\xi \sim P}[\cdot]$ in both sides of the result in Lemma 8 and interchanging the expectations. The interchange can be done using Fubini's Theorem since for every $x \in X$ and for every $z \in Z$ we have that $\mathcal{L}_t(x, z) < O(\frac{1}{\alpha})$ almost surely. □

We are now ready to prove Theorem 3. We assume f_t is 1-Lipschitz continuous.

Proof of Theorem 3. Define concentration error $CE = C_\alpha[\{f_t(x^*)\}_{t=1}^T] - C_\alpha[\{f_t(\bar{x})\}_{t=1}^T]$, where $\bar{x} = \arg\min_{x \in X} C_\alpha[\{f_t(x)\}_{t=1}^T]$, let $x^* = \arg\min_{x \in X} C_\alpha[F](x)$, we have

$$\begin{aligned}
&\mathbb{E}[C_\alpha[\{f_t(x_t)\}_{t=1}^T] \pm C_\alpha[\{f_t(x^*)\}_{t=1}^T]] - min_{x \in X} C_\alpha[\{f_t(x)\}_{t=1}^T] \\
&= \mathbb{E}[\min_y y + \frac{1}{\alpha T}\sum_{t=1}^T \max\{f_t(x_t) + f_t(x^*) - f_t(x^*) - y, 0\} - C_\alpha[\{f_t(x^*)\}_{t=1}^T]] \\
&\quad + \mathbb{E}[CE] \\
&\leq \mathbb{E}[\min_y y + \frac{1}{\alpha T}\sum_{t=1}^T \max\{f_t(x^*) + |f_t(x_t) - f_t(x^*)| - y, 0\} - C_\alpha[\{f_t(x^*)\}_{t=1}^T]] \\
&\quad + \mathbb{E}[CE] \\
&\leq \mathbb{E}[\min_y y + \frac{1}{\alpha T}\sum_{t=1}^T \max\{f_t(x^*) + |f_t(x_t) - f_t(x^*)| - y, |f_t(x_t) - f_t(x^*)|\} \\
&\quad - C_\alpha[\{f_t(x^*)\}_{t=1}^T]] + \mathbb{E}[CE]
\end{aligned}$$

$$= \mathbb{E}[\min_y y + \frac{1}{\alpha T}\sum_{t=1}^{T}\max\{f_t(x^*) - y, 0\} + \frac{1}{\alpha T}\sum_{t=1}^{T} + |f_t(x_t) - f_t(x^*)|$$

$$- C_\alpha[\{f_t(x^*)\}_{t=1}^T]] + \mathbb{E}[CE]$$

$$= \mathbb{E}[\frac{1}{\alpha T}\sum_{t=1}^{T}|f_t(x_t) - f(x^*)|] + \mathbb{E}[CE]$$

$$\leq \frac{1}{\alpha T}\sum_{t=1}^{T}\mathbb{E}_t[||x_t - x^*||] + \mathbb{E}[CE] \quad \text{since } f_t \text{ is 1-Lipschitz}$$

$$\leq \frac{1}{\alpha T}\sqrt{T}\sqrt{\sum_{t=1}^{T}\mathbb{E}_t[||x_t - x^*||]^2} + \mathbb{E}[CE] \quad \text{by Cauchy Schwartz}$$

$$\leq \frac{1}{\alpha T}\sqrt{T}\sqrt{\sum_{t=1}^{T}\mathbb{E}_t[\frac{2}{\beta}[C_\alpha[F](x_t) - C_\alpha[F](x^*)]]} + \mathbb{E}[CE]$$

by strong convexity of $C_\alpha[F](\cdot)$ and KKT condition

$$= \frac{1}{\alpha T}\sqrt{T}\sqrt{\frac{2}{\beta}\mathbb{E}[\sum_{t=1}^{T}C_\alpha[F](x_t) - C_\alpha[F](x^*)]} + \mathbb{E}[CE]$$

$$= O(\frac{d^{1/2}}{\alpha^{3/2}\beta^{1/2}T^{1/8}}) + \mathbb{E}[CE] \quad \text{by Theorem 2}$$

We still need to bound the concentration error CE in expectation. Notice we can write

$$CE = [C_\alpha[\{f_t(x^*)\}_{t=1}^T] - C_\alpha[F](x^*)] + [C_\alpha[F](x^*) - C_\alpha[F](\bar{x})]$$
$$+ [C_\alpha[F](\bar{x}) - C_\alpha[\{f_t(\bar{x})\}_{t=1}^T]$$

and the second term is nonpositive. To bound CE in expectation we apply Lemma 65 on functions $\phi(x, y) = y + \frac{1}{\alpha}[f(x) - y]_+$ (notice $L \leq O(\frac{1}{\alpha})$ and $R = O(\frac{1}{\alpha})$), by Remark 1 and the same reasoning as in the proof of Lemma 65 we have $\mathbb{E}[|C_\alpha[F](\bar{x}) - C_\alpha[\{f_t(\bar{x})\}_{t=1}^T|] \leq \tilde{O}(\frac{\sqrt{d}}{\alpha\sqrt{T}})$. Thus $\mathbb{E}[CE] \leq \tilde{O}(\frac{\sqrt{d}}{\alpha\sqrt{T}})$. This finishes the proof. □

2.7 Algorithm 2

Algorithm 1, while it is intuitive and easy to implement, does not achieve the optimal pseudo-regret bound of $\frac{1}{\sqrt{T}}$. In this section, we adapt an algorithm from [9] that achieves the optimal regret bound (up to logarithmic factors), unfortunately its dependency on d is less than ideal. We consider the cases $d = 1$ and $d > 1$ separately.

2.7.1 The 1-Dimensional Case

For simplicity, in this section we assume that $X = [0, 1]$ and that $f(\cdot, \xi)$ is 1-Lipschitz continuous for every $\xi \in \Xi$. This implies that $C_\alpha[F](\cdot)$ is also 1-Lipschitz continuous (see Lemma 66 in the appendix). We let $LB_{\gamma_i}(x)$ and $UB_{\gamma_i}(x)$ denote the $C_\alpha[F](\cdot)$ lower and upper bounds of the confidence intervals (CI's) of width γ_i at point x.

The algorithm proceeds in epochs and rounds. In epoch τ the algorithm works with region $[l_\tau, r_\tau]$. In this region we will be playing three points x_l, x_c, x_r (x_c is the center point) for several rounds $i = 1, 2, \ldots$. In each round i the algorithm will play $\frac{\ln(T/(\alpha\gamma))}{\alpha^2\gamma_i^2}$ times the aforementioned points and build CI's for $C_\alpha[F]$. Roughly speaking, the reason why the algorithm works is because in every round we are 1) either playing points such that we are not suffering too much pseudo-regret or 2) we are quickly identifying a subregion of the working region which only contains "bad points" and discarding it. Every time 2) occurs we are shrinking the working region by a constant factor, this will guarantee that after not too many rounds we are only working with a small feasible region.

For convenience we denote $h(x) := C_\alpha[F](x)$ and $x^* := \text{argmin}_{x \in X} h(x)$. Notice that the minimizer need not be unique in which case we choose one arbitrarily. At the end of a round one of the following occurs:

Case 1. The CI's around $h(x_l)$ and $h(x_r)$ are sufficiently separated. If this is the case, then by convexity we can discard one fourth of the working feasible region: either the one to the left of x_l or the one to the right of x_r .

Algorithm 2 $(d = 1)$

Input: Input: $X \in [0, 1]$, total number of time-steps T
Let $l_1 := 0, r_1 := 1$
for epoch $\tau = 1, 2, ...$ **do**
  Let $w_\tau := r_\tau - l_\tau$
  Let $x_l := l_\tau + w_\tau/4, x_c = l_\tau + w_\tau/2, x_r := l_\tau + 3w_\tau/4$
  for round $i = 1, 2, ...$: **do**
    Let $\gamma_i = 2^{-i}$
    For each $x \in \{x_l, x_c, x_r\}$ play x $\frac{\ln(T/(\alpha\gamma))}{\gamma_i^2\alpha^2}$ times and build CI's: $[\hat{C}_\alpha[F](x_k)] - \gamma_i, \hat{C}_\alpha[F](x_k) + \gamma_i]$ for $k \in \{l, c, r\}$
    if $\max\{LB_{\gamma_i}(x_l), LB_{\gamma_i}(x_r)\} \geq \min\{UB_{\gamma_i}(x_l), UB_{\gamma_i}(x_r)\} + \gamma_i$ (Case 1) **then**
      if $LB_{\gamma_i}(x_l) \geq LB_{\gamma_i}(x_r)$ **then**
        set $l_{\tau+1} := x_l$ and $r_{\tau+1} := r_\tau$
      else
        set $l_{\tau+1} := l_\tau$ and $r_{\tau+1} := x_r$
      end if
      Continue to epoch $\tau + 1$
    else if $\max\{LB_{\gamma_i}(x_l), LB_{\gamma_i}(x_r)\} \geq UB_{\gamma_i}(x_c) + \gamma_i$ (Case 2) **then**
      if $LB_{\gamma_i}(x_l) \geq LB_{\gamma_i}(x_r)$ **then**
        set $l_{\tau+1} := x_l$ and $r_{\tau+1} := r_\tau$
      else
        set $l_{\tau+1} := l_\tau$ and $r_{\tau+1} := x_r$
      end if
      Continue to epoch $\tau + 1$
    end if (Case 3)
  end for
end for

Case 2. If Case 1 does not occur, the algorithm checks if the CI around $h(x_c)$ is sufficiently below at least one of the CI's around $h(x_l)$ or $h(x_r)$. If this is the case then we can discard one fourth of the working feasible region.

Case 3. If neither Case 1 or Case 2 occurs then we can be sure that the function is flat in the working feasible region (as measured by γ) and thus we are not incurring a very high pseudo-regret.

The main results of this section are the following.

Theorem 4. *With probability at least* $1 - \frac{1}{T}$*, Algorithm 2 (1-D) guarantees*

$$\bar{\mathcal{R}}_T \leq \frac{\kappa \ln(T)}{\sqrt{T}\alpha} \ln(\frac{\alpha T}{\ln(T)}).$$

Theorem 5. *Let* $f(\cdot, \xi)$ *be strongly convex with parameter* $\beta > 0$ *for all* $\xi \in \Xi$*. With probability at least* $1 - \frac{3}{T}$*, Algorithm 2 (1-D) guarantees*

$$\mathcal{R}_T \leq \tilde{O}(\frac{1}{\alpha^{3/2}\beta^{1/2}T^{1/4}}).$$

We follow [9] for the analysis of the algorithm. The main difference in the analysis is that we must build estimates of the $CVaR$ of the random loss at every point instead of building them for the expected loss. Because of this, we have to use different concentration results which directly affect how many times we must choose an action.The detailed analysis of the algorithm and the proofs of the theorems in this section can be found in the next subsection.

2.7.2 Analysis of Algorithm 2 (1-D)

We proceed to formally analyze the algorithm following [9]. In this section, for ease of reading we refer to quantity $T\bar{\mathcal{R}}_T$ as the regret. We work conditioned on $\mathcal{E}$ which is defined as the event that for every epoch and for every round i, $h(x) \in [LB_{\gamma_i}(x), UB_{\gamma_i}(x)]$ for $x \in \{x_l, x_c, x_r\}$. We will first bound the regret in an epoch and then bound the total number of epochs. We do the previous in the next sequence of lemmas. Notice that by Lemma 1 we can obtain a γ-CI for $h(x)$ that holds with probability at least $1 - \frac{1}{T^2}$ with only $\frac{\kappa \ln(T/(\alpha\gamma))}{\alpha^2\gamma^2}$ samples. We first show that we never discard points that are near optimal.

Lemma 9. *If epoch* τ *ends in round* i*, then the interval* $[l_{\tau+1}, r_{\tau+1}]$ *contains every* $x \in [l_\tau, r_\tau]$ *such that* $h(x) \leq h(x^*) + \gamma_i$*. In particular,* $x^* \in [l_\tau, r_\tau]$ *for all epochs* τ*.*

Proof. Assume epoch τ terminates in round i through Case 1. Then, either $LB_{\gamma_i}(x_l) \geq$

$UB_{\gamma_i}(x_r) + \gamma_i$ or $LB_{\gamma_i}(x_r) \geq UB_{\gamma_i}(x_l) + \gamma_i$. We assume the former occurs. It then holds that

$$h(x_l) \geq h(x_r) + \gamma_i.$$

We must show that the points in the working feasible region to the left of x_l are not near optimal. That is, for every $x \in [l_\tau, l_{\tau+1}] = [l_\tau, x_l]$ we have $h(x) \geq h(x^*) + \gamma_i$. Pick $x \in [l_\tau, x_l]$ then, for some $t \in [0, 1]$ we have $x_l = tx + (1-t)x_r$. Since h is convex we have

$$h(x_l) \leq th(x) + (1-t)h(x_r)$$

which implies

$$\begin{aligned} h(x) &\geq h(x_r) + \frac{h(x_l) - h(x_r)}{t} \\ &\geq h(x_r) + \frac{\gamma_i}{t} \\ &\geq h(x^*) + \gamma_i \end{aligned}$$

as required. If $LB_{\gamma_i}(x_r) \geq UB_{\gamma_i}(x_l) + \gamma_i$ had occurred the argument is analogous.

If epoch τ had terminated through case 2 then

$$\max\{LB_{\gamma_i}(x_l), LB_{\gamma_i}(x_r)\} \geq UB_{\gamma_i}(x_c) + \gamma_i.$$

We assume $LB_{\gamma_i}(x_l) \geq UB_{\gamma_i}(x_c) + \gamma_i$, then

$$h(x_l) \geq h(x_c) + \gamma_i.$$

The same argument as above with x_c instead of x_r guarantees $h(x_l) \geq h(x^*) + \gamma_i$. If

$LB_{\gamma_i}(x_r) \geq UB_{\gamma_i}(x_c) + \gamma_i$ had occurred the argument is analogous. The fact that $x^* \in [l_\tau, r_\tau]$ for every epoch τ follows by induction. $\square$

We now show that if an epoch does not terminate in a given round i then the regret ($T\bar{\mathcal{R}}_T$) incurred in that epoch was not too high.

Lemma 10. *If epoch τ continues from round i to $i+1$ then the regret in round i is at most*

$$\frac{\kappa \ln(T/(\alpha\gamma_i))}{\alpha^2\gamma_i}$$

Proof. The regret incurred in round i of epoch τ is

$$\frac{\kappa \ln(T/(\alpha\gamma_i))}{\alpha^2\gamma_i^2}[(h(x_l) - h(x^*)) + (h(x_c) - h(x^*)) + (h(x_r) - h(x^*))]$$

It suffices to show that for every $x \in \{x_l, x_c, x_r\}$ it holds that

$$h(x) \leq h(x^*) + 12\gamma_i.$$

The algorithm continues from round i to round $i+1$ if and only if

$$\max\{LB_{\gamma_i}(x_l), LB_{\gamma_i}(x_r)\} < \min\{UB_{\gamma_i}(x_l), UB_{\gamma_i}(x_r)\} + \gamma_i$$

and

$$\max\{LB_{\gamma_i}(x_l), LB_{\gamma_i}(x_r)\} < UB_{\gamma_i}(x_c) + \gamma_i.$$

This implies that $h(x_l), h(x_c)$, and $h(x_r)$ are all contained in an interval of at most $3\gamma_i$. There are two cases for which the argument is essentially the same, either $x^* \leq x_c$ or $x^* > x_c$, we consider the former. Since by the previous lemma we know that $x^* \in [l_\tau, r_\tau]$,

then there exists $t \in [0,1]$ such that $x^* = x_c + t(x_c - x_r)$. Therefore

$$x_c = \frac{1}{1+t}x^* + \frac{t}{1+t}x_r.$$

Since $|x_c - l_\tau| = w_\tau/2$ and $|x_r - x_c| = w_\tau/4$ we have

$$t = \frac{|x^* - x_c|}{|x_r - x_c|} \leq \frac{|l_\tau - x_c|}{|x_r - x_c|} = \frac{w_\tau/2}{w_\tau/4} = 2$$

Since h is convex

$$h(x_c) \leq \frac{1}{1+t}h(x^*) + \frac{t}{1+t}h(x_r)$$

therefore

$$\begin{aligned}
h(x^*) &\geq (1+t)\big(h(x_c) - \frac{t}{1+t}h(x_r)\big) \\
&= h(x_c) + (1+t)(h(x_c) - h(x_r)) \\
&\geq h(x_c) - (1+t)|h(x_c) - h(x_r)| \\
&\geq h(x_r) - (1+t)3\gamma_i \\
&\geq h(x_r) - 9\gamma_i
\end{aligned}$$

So, for all $x \in \{x_l, x_c, x_r\}$ it holds that

$$h(x) \leq h(x_r) + 3\gamma_i \leq h(x^*) + 12\gamma_i.$$

□

We proceed to bound the regret in each epoch.

Lemma 11. *If epoch τ ends in round i the regret incurred in the epoch is no more than*

$$\frac{\kappa \ln(T/(\alpha\gamma_i))}{\alpha^2\gamma_i}.$$

Proof. If $i = 1$, since $h(x)$ is 1-Lipschitz and $X = [0, 1]$ we have that for every $x \in \{x_l, x_c, x_r\}$ $h(x) - h(x^*) \leq 1$. Therefore the regret in epoch τ is

$$\begin{aligned}
&\frac{\kappa \ln(T/(\alpha^2\gamma_i^2))}{\alpha^2\gamma_i^2}\big(((h(x_l) - h(x^*)) + (h(x_c) - h(x^*)) + (h(x_r) - h(x^*))\big) \\
&\leq \frac{6\kappa \ln(T/(\alpha^2\gamma_i^2))}{\alpha^2\gamma_1}
\end{aligned}$$

If $i \geq 2$, by the previous lemma we have that the regret incurred in round j with $1 \leq j \leq i - 1$ is no more than

$$\frac{\kappa \ln(T/(\alpha^2\gamma_i^2))}{\alpha^2\gamma_j}.$$

For round i the regret incurred is at most

$$3 \cdot 12\gamma_{i-1}\frac{\kappa \ln(T/(\alpha^2\gamma_i^2))}{\alpha^2\gamma_i^2} = \frac{\kappa 72 \ln(T/(\alpha^2\gamma_i^2))}{\alpha^2\gamma_i}.$$

It follows that the regret in epoch τ is

$$\begin{aligned}
&\sum_{j=1}^{i-1} \frac{\kappa \ln(T/(\alpha^2\gamma_j^2))}{\alpha^2\gamma_j} + \frac{\kappa \ln(T/(\alpha^2\gamma_i^2))}{\alpha^2\gamma_i} \\
&= \sum_{j=1}^{i-1} \frac{\kappa \ln(T/(\alpha^2\gamma_j^2))}{\alpha^2} \cdot 2^j + \frac{\kappa \ln(T/(\alpha^2\gamma_i^2))}{\alpha^2\gamma_i} \\
&< \frac{\kappa \ln(T/(\alpha^2\gamma_i^2))}{\alpha^2} \cdot 2^i + \frac{\kappa \ln(T/(\alpha^2\gamma_i^2))}{\alpha^2\gamma_i} \\
&= \frac{\kappa \ln(T/(\alpha\gamma_i))}{\alpha^2\gamma_i}.
\end{aligned}$$

□

We have bounded the regret that we incur in each epoch. We proceed to bound the number of epochs.

Lemma 12. *The total number of epochs τ satisfies*

$$\tau \leq \kappa \log_{4/3}(\frac{\alpha^2 T}{\ln(T)}).$$

Proof. The key is to observe that since the number of times we sample a point is bounded above by T then $\gamma_i \geq (\alpha^2 T/(\kappa \ln(T)))^{-1/2}$ for every round and every epoch. Let $\gamma_{min} := (\alpha^2 T/(\kappa \ln(T)))^{-1/2}$ and let $I := [x^* - \gamma_{min}, x^* + \gamma_{min}]$. Since h is 1-Lipschitz, for any $x \in I$

$$h(x) - h(x^*) \leq \gamma_{min}.$$

By Lemma 9 we have that for any round τ' which ends in round i'

$$I \subseteq \{x \in [0,1] : f(x) < f(x^*) + \gamma_{i'}\} \subseteq [l_{\tau'+1}, r_{\tau'+1}]$$

since $\gamma_{min} \leq \gamma_{i'}$. The previous implies

$$2\gamma_{min} \leq r_{\tau+1} - l_{\tau+1} = w_{\tau+1}.$$

By the definitions of $l_{\tau'+1}$, $r_{\tau'+1}$ and $w_{\tau'+1}$ we have that for any $\tau' \in \{1, ..., \tau\}$

$$w_{\tau'+1} \leq \frac{3}{4} w_{\tau'}.$$

Therefore,

$$2\gamma_{min} \leq w_{\tau+1} \leq (\frac{3}{4})^{\tau} w_1 \leq (\frac{3}{4})^{\tau}$$

which yields the result. □

We are now ready to prove Theorems 4 and 5 .

Proof of Theorem 4. The per epoch regret when epoch τ ends in round i is

$$\frac{\kappa \ln(T/(\alpha\gamma_i))}{\alpha^2\gamma_i} \leq \frac{\kappa \ln(T/(\alpha\gamma_i))}{\alpha^2\gamma_{\min}} \leq \frac{\kappa\sqrt{T}\ln(T/(\alpha\gamma_{\min}))}{\alpha} = \frac{\kappa\sqrt{T}\ln(T)}{\alpha}.$$

Using the previous lemma we know that the regret will not be more than

$$\frac{\kappa\sqrt{T}\ln(T)}{\alpha}\log_{4/3}(\frac{\alpha^2 T}{\ln(T)})$$

Recall we have been working conditioned on $\mathcal{E}$. We need an upper bound on $P(\mathcal{E}')$. We know that after $\frac{\kappa \ln(T/(\alpha\gamma))}{\alpha^2\gamma_i}$ queries we have

$$P(|\hat{h}(x) - h(x)| \geq \gamma_i) \leq \frac{1}{T^2}.$$

Since there are at most T epochs a union bound gives

$$P(\mathcal{E}') \leq \frac{1}{T}$$

which yields the desired result. □

Proof of Theorem 5. The proof is very similar to that of Theorem 3 with the difference that we have to bound the concentration error $CE := C_\alpha[\{f_t(x^*)\}_{t=1}^T] - \min_{x\in X} C_\alpha[\{f_t(x)\}_{t=1}^T]$ with high probability. As explained in the proof of Theorem 3 we know

$$CE \leq |C_\alpha[\{f_t(x^*)\}_{t=1}^T] - C_\alpha[F](x^*)| + |C_\alpha[F](\bar{x}) - C_\alpha[\{f_t(\bar{x})\}_{t=1}^T]|$$

where $\bar{x} = \arg\min_{x\in X} C_\alpha[\{f_t(x)\}_{t=1}^T]$. To bound CE with high probability we apply Lemma 64 with $\delta = 1/T$ on functions $\phi(x,y) = y + \frac{1}{\alpha}[f(x) - y]_+$ (notice $L \leq O(\frac{1}{\alpha})$

and $R = O(\frac{1}{\alpha})$), by the same reasoning as in the proof of Lemma 1 we have that with probability at least $1 - \frac{1}{T}$, $|C_\alpha[F](\bar{x}) - C_\alpha[\{f_t(\bar{x})\}_{t=1}^T]| \leq \tilde{O}(\frac{1}{\alpha\sqrt{T}})$ and thus by a union bound we have that with probability at least $1 - \frac{2}{T}$, $CE \leq \tilde{O}(\frac{1}{\alpha\sqrt{T}})$. As in the proof of Theorem 3 we have

$$\mathcal{R}_T \leq \frac{\sqrt{T}}{\alpha T \beta^{1/2}} \sqrt{T\bar{\mathcal{R}}_T} + CE.$$

Using Theorem 4 to bound $\bar{\mathcal{R}}_T$, the argument in the previous paragraph to bound CE, and a union bound yields the result. □

2.7.3 The d-Dimensional Case

Next, we give some intuition on how the algorithm works. Let us first consider the problem of minimizing a convex function over a bounded set with a first-order oracle (i.e. a gradient and function value oracle). For simplicity let us assume that the convex set is a ball. An ellipsoid-type method would work really well in this setup because of the following. By querying the first order oracle at any point (due to convexity) we could identify a subregion of the current feasible region where the function value is worse than the function value at the point we made the query. If we could somehow discard that bad portion of the feasible set, and the size of this bad region is big enough, by iterating the procedure (assuming this can be done) we should end up with a set that only has points close to optimal.

Let us now consider a similar but harder problem of minimizing a convex function over a a bounded set (say a ball) with a zeroth-order oracle (i.e. a function value oracle). In this setup, with one query, we can no longer identify a subregion of the current feasible region where the function values are worse than the function value at the point we made the query. A first approach to tackle this problem is the following. Build a small regular simplex centered at the origin of the ball and query the function at its vertices. Assume the maximal function value occurs at vertex y', then by convexity of the function one can

conclude that the cone generated by reflecting the simplex around y' is a region where the function values are bad. Since we have identified a bad region of the feasible set we would like to discard it and keep iterating our method, unfortunately what remains of the ball when we discard the cone is a non-convex set we can not keep iterating the method. To try to fix the previous one could try to find the minimum volume enclosing ellipsoid of the non-convex set and keep iterating. Unfortunately this does not work since the minimum volume enclosing ellipsoid will not have sufficiently small volume [103]. The reason this occurs is that the angle of the cone generated by reflecting the simplex around y' is not wide enough. In [103] the authors fix the previous by constructing a pyramid (with wide enough angle) with y' as its apex and sample the vertices of the pyramid. If we are lucky enough and y' has the maximal function value among all the vertices of the pyramid, we can then discard the cone generated by reflecting the pyramid around y' and enclose that region in the minimum volume ellipsoid. However, if we were not lucky enough and y' did not have the maximal function value then, Nemirovski and Yudin [103], show that by repeatedly building a new pyramid with apex at the point with maximal function value we will identify a bad region after building not too many pyramids. It is not to hard to see that the previous approach may work even if we have a noisy-zeroth-order oracle, as long as the noise is not too large. The previous approach describes an optimization procedure but by itself it does not guarantee low regret. However, by incorporating center points, sublinear regret can be achieved. For clarity reasons the algorithm and its analysis are presented in the next subsection. The main results from this section are the following.

Theorem 6. *Algorithm 2 run with parameters* $c_1 \geq 64, c_2 \leq 1/32$ *and*

$$\Delta_\tau(\gamma) = \big(\frac{6c_1 d^4}{c_2^2} + 3\big)\gamma, \quad \bar{\Delta}_\tau(\gamma) = \big(\frac{6c_1 d^4}{c_2^2} + 5\big)\gamma,$$

guarantees that with probability at least $1 - \frac{1}{T}$

$$\bar{\mathcal{R}}_T \leq \frac{\kappa d^3 \ln(T/\alpha) \ln(T)}{\sqrt{T}\alpha^2} \big(\frac{2d^2 \ln(d)}{c_2^2} + 1 \big) \big(\frac{4d^7 c_1}{c_2^3} + \frac{d(d+2)}{c_2} \big) \big(\frac{12 c_1 d^4}{c_2^2} + 11 \big).$$

Theorem 7. *Let* $f(\cdot, \xi)$ *be strongly convex with parameter* $\beta > 0$ *for any* $\xi \in \Xi$*, Algorithm 2 run with the same parameters as in Theorem 6 guarantees that with probability at least* $1 - \frac{3}{T}$

$$\mathcal{R}_T \leq \tilde{O}(\frac{d^8}{\alpha^3 \beta^{1/2} T^{1/4}}).$$

2.7.4 Analysis of Algorithm 2 (d-D)

We are now ready to describe the algorithm informally. As in the special case from the previous section, Algorithm 2 proceeds in epochs. Let the initial working feasible region be $\mathcal{X}_0 = X$. The goal is that at the end of every epoch τ we will discard some portion of the working region $\mathcal{X}_\tau$ and end up with a smaller region $\mathcal{X}_{\tau+1}$ which contains at least one approximate optimum.

We now give a brief description of the algorithm. At the beginning of every epoch τ we apply an affine transformation to the current working region $\mathcal{X}_\tau$ such that the smallest ellipsoid that contains it is an Euclidean ball of radius R_τ which we denote $\mathcal{B}(R\tau)$. We assume that $R_1 \leq 1$. Let $r_\tau := R_\tau/(c_1 d)$ for some $c_1 \geq 1$ so that $\mathcal{B}(r_\tau) \subseteq \mathcal{X}_\tau$ (such a construction is always possible see Lecture 1 p. 2 of [21]). We refer to the enclosing ball $\mathcal{B}(R_\tau)$ as $\mathcal{B}_\tau$. Every epoch will consist of several rounds where γ_i is halved in every round. Let x_0 be the center of $\mathcal{B}_\tau$. At the start of epoch τ, we build a simplex with center x_0 contained in $\mathcal{B}(r_\tau)$. We will play the vertices of the simplex $x_1, ..., x_{d+1}$ enough times so that the CI's at each vertex are of width γ_i and hold with high probability. The algorithm will then choose point y_1 for which $\hat{h}(x)_i$ is the largest, here $\hat{h}$ denotes the empirical estimate of h. By construction we are guaranteed that $h(y_1) \geq h(x_j) - \gamma_i$ for $j = 1, ..., d+1$.

The algorithm will now try to identify a region where the function value is high so that at the end of the epoch we can discard it. It will do this by constructing pyramids with parameter $\hat{\gamma}$ (always greater that γ) until a bad region is found, if this does not happen for the current value of γ it means that the algorithm did not incur to much regret (relative to how large γ was). The pyramid construction follows from Section 9.2.2 of [103]. The pyramids have angle 2ϕ at the apex where $\cos(\phi) = c_2/d$. The base of the pyramid has d vertices, $z_1, ..., z_d$ such that $z_i - x_0$ and $y_1 - z_i$ are orthogonal. The previous construction is always possible. Indeed, take a sphere with diameter $y_1 - x_0$ and arrange $z_1, ...z_d$ on its boundary such that the angle between $y_1 - x_0$ and $y_1 - z_i$ is ϕ. We now set $\hat{\gamma} = 1$ and play all the points $y_1, z_1, ...z_d$, and the center of the pyramid enough times until all the CI's are of width $\hat{\gamma}$. Let TOP and BOTTOM be the vertices of the pyramid (including y_1) with the largest and smallest values for $\hat{h}(x)$. Let $\Delta(\cdot), \bar{\Delta}(\cdot)$, be functions which are specified later. We then check for one of the following cases:

1. If $LB_{\hat{\gamma}}(\text{TOP}) \geq UB_{\hat{\gamma}}(\text{BOTTOM}) + \Delta_\tau(\hat{\gamma})$ then we proceed depending on what the separation between the CI's of TOP and APEX is.

   (a) If $LB_{\hat{\gamma}}(\text{TOP}) \geq UB_{\hat{\gamma}}(\text{APEX}) + \hat{\gamma}$, then with high probability

   $$h(\text{TOP}) \geq h(\text{APEX}) + \hat{\gamma} \geq h(\text{APEX}) + \gamma_i.$$

   We then build a new pyramid with apex equal to TOP, reset $\hat{\gamma} = 1$ and continue sampling on the new pyramid.

   (b) If $LB_{\hat{\gamma}}(\text{TOP}) < UB_{\hat{\gamma}}(\text{APEX}) + \hat{\gamma}$, then $LB_{\hat{\gamma}}(\text{APEX}) \geq UB_{\hat{\gamma}}(\text{BOTTOM}) + \Delta(\hat{\gamma}) - 2\hat{\gamma}$. We then conclude the epoch and pass the current apex to the cone-cutting subroutine.

2. If $LB_{\hat{\gamma}}(\text{TOP}) < UB_{\hat{\gamma}}(\text{BOTTOM}) + \Delta_\tau(\hat{\gamma})$, then one of the following things happen:

   (a) If $UB_{\hat{\gamma}}(\text{CENTER}) \geq LB_{\hat{\gamma}}(\text{BOTTOM}) - \bar{\Delta}_\tau(\hat{\gamma})$, then all the vertices of the pyra-

mid and the center of the pyramid have function values in an interval of size $2\Delta_\tau(\hat{\gamma}) + 3\hat{\gamma}$. We can then set $\hat{\gamma} = \hat{\gamma}/2$. If $\hat{\gamma} < \gamma_i$, we start the next round with $\gamma_{i+1} = \gamma_i/2$. Otherwise we continue sampling with the new $\hat{\gamma}$.

(b) If $UB_{\hat{\gamma}}(\text{CENTER}) < LB_{\hat{\gamma}}(\text{BOTTOM}) - \bar{\Delta}_\tau(\hat{\gamma})$. We conclude the epoch and pass the center and current apex to the hat-raising subroutine.

Hat-Raising: This occurs whenever the pyramid satisfies $LB_{\hat{\gamma}}(\text{TOP}) \leq UB_{\hat{\gamma}}(\text{BOTTOM}) + \Delta_\tau(\hat{\gamma})$ and $UB_{\hat{\gamma}}(\text{CENT}) \leq LB_{\hat{\gamma}}(\text{BOTTOM}) - \bar{\Delta}_\tau(\hat{\gamma})$. We will later show that if we move the apex a little from y_i to y_i', then the CI of y_i' is above the CI of TOP and the new angle ϕ' in not too much smaller than 2ϕ. In particular, we will let $y_i' = y_i + (y_i - \text{CENTER}_i)$.

Cone-cutting: This is the last step in a given epoch (notice this is the last step in the hat-raising subroutine). This subroutine receives a pyramid with apex y and base $z_1, ..., z_d$ with angle $2\bar{\phi}$ at the apex such that $\cos(\bar{\phi}) \leq 1/2d$. Define the cone

$$K_\tau = \{x : \exists \lambda > 0, \alpha_1, ..., \alpha_d > 0, \sum_{i=1}^{d} \alpha_i = 1 : x = y - \lambda \sum_{i=1}^{d} \alpha_i (z_i - y)\} \tag{2.9}$$

which is centered at y and is the reflection of the pyramid around the apex. By construction $\mathcal{K}_\tau$ has angle $2\bar{\phi}$ at the apex. Let $\mathcal{B}'_{\tau+1}$ be the minimum volume ellipsoid that contains $\mathcal{B}_\tau \setminus \mathcal{K}_\tau$ and let $\mathcal{X}_{\tau+1} = \mathcal{X}_\tau \cap \mathcal{B}'_{\tau+1}$. Finally, by applying an affine transformation to $\mathcal{B}'_{\tau+1}$ we obtain $\mathcal{B}_{\tau+1}$.

Before proving that the algorithm achieves low regret we discuss the computational aspects of the algorithm. The most computationally intensive steps are cone-cutting, and the isotropic transformation that transforms $B'_{\tau+1}$ into a sphere $B_{\tau+1}$. These steps are analogous to the implementation of the ellipsoid algorithm. In particular, there is an equation for $B'_{\tau+1}$ see [61]. The affine transformations can be computed via rank one matrix updates and therefore the computation of inverses can be done efficiently.

We follow [9] for the analysis of the algorithm. The main difference in the analysis

Algorithm 2 ($X \subset \mathbb{R}^d$)

Input: X, constants c_1 and c_2, functions $\Delta_\tau(\gamma)$ and $\hat{\Delta}_\tau(\gamma)$, and total number of time-steps T

Let $\mathcal{X}_1 = X$

for epoch $\tau = 1, 2, ...$ **do**

Round $\mathcal{X}_t$ so $\mathcal{B}(r_\tau) \subseteq \mathcal{X}_\tau \subseteq \mathcal{R}(R_\tau)$, R_τ is minimized and $r_\tau := R_\tau/(c_1 d)$. Let $\mathcal{B}_\tau = \mathcal{B}(R_\tau)$.

Build a simplex with vertices $x_1, ..., x_{d+1}$ on the surface of $\mathcal{B}(r_\tau)$.

for round $i = 1, 2, ...$ **do**

Let $\gamma_i := 2^{-i}$

Play x_j for each $j = 1, ..., d+1$, $\kappa \frac{\ln(T/(\alpha\gamma))}{\alpha^2 \gamma_i^2}$ times and build CI's: $[\hat{C}_\alpha[F](x_j) - \gamma_i, \hat{C}_\alpha[F](x_j) + \gamma_i]$

Let $y_1 := \arg\max_{x_j} LB_{\gamma_i}(x_j)$

for pyramid $k = 1, 2, ...$ **do**

Construct pyramid Π_k with apex y_k; let $z_1, ..., z_d$ be the vertices of the base of Π_k and z_0 be the center of Π_k

loop

Play each of $\{y_k, z_0, z_1, ..., z_d\}$, $\kappa \frac{\ln(T/(\alpha\gamma))}{\alpha^2 \gamma_i^2}$ times and build CI's

Let CENTER $:= z_0$, APEX $:= y_k$, TOP be the vertex v of Π_k maximizing $LB_{\hat{\gamma}}(v)$, BOTTOM be the vertex v of Π_k minimizing $LB_{\hat{\gamma}}(v)$

if $LB_{\hat{\gamma}}(\text{TOP}) \geq UB_{\hat{\gamma}}(\text{BOT}) + \Delta_\tau(\hat{\gamma})$ and $LB_{\hat{\gamma}}(\text{TOP}) \geq UB_{\hat{\gamma}}(\text{APEX}) + \hat{\gamma}$: (Case 1a)) **then**

Let $y_{k+1} :=$ TOP, immediately continue to pyramid $k+1$

else if $LB_{\hat{\gamma}}(\text{TOP}) \geq UB_{\hat{\gamma}}(\text{BOT}) + \Delta_\tau(\hat{\gamma})$ and $LB_{\hat{\gamma}}(\text{TOP}) < UB_{\hat{\gamma}}(\text{APEX}) + \hat{\gamma}$: (Case 1b)) **then**

Set $(\mathcal{X}_{\tau+1}, \mathcal{B}_{\tau+1}) =$ CONE-CUTTING$(\Pi_k, \mathcal{X}_\tau, \mathcal{B}_\tau)$, proceed to epoch $\tau + 1$

else if $LB_{\hat{\gamma}}(\text{TOP}) < UB_{\hat{\gamma}}(\text{BOT}) + \Delta_\tau(\hat{\gamma})$ and $UB_{\hat{\gamma}}(\text{CENT}) \geq LB_{\hat{\gamma}}(\text{BOT}) - \bar{\Delta}_\tau(\hat{\gamma})$: (Case 2a)) **then**

Let $\hat{\gamma} := \hat{\gamma}/2$

if $\hat{\gamma} < \gamma_i$ **then**

Start next round $i+1$

end if

else if $LB_{\hat{\gamma}}(\text{TOP}) < UB_{\hat{\gamma}}(\text{BOT}) + \Delta_\tau(\hat{\gamma})$ and $UB_{\hat{\gamma}}(\text{CENT}) < LB_{\hat{\gamma}}(\text{BOT}) - \bar{\Delta}_\tau(\hat{\gamma})$: (Case 2b)) **then**

Set $(\mathcal{X}_{\tau+1}, \mathcal{B}_{\tau+1}) =$ HAT-RAISING$(\Pi_k, \mathcal{X}_\tau, \mathcal{B}_\tau)$ and proceed to epoch $\tau + 1$

end if

end loop

end for

end for

end for

Algorithm CONE-CUTTING

Input: pyramid Π with apex y, (rounded) feasible region $\mathcal{X}_\tau$ for each epoch τ, enclosing ball $\mathcal{B}_\tau$

1. Let $z_1, ..., z_d$ be the vertices of the base of Π, and ϕ the angle at its apex.
2. Define the cone $\mathcal{K}_\tau = \{x | \exists \lambda > 0, \alpha_1, ..., \alpha_d > 0, \sum_{i=1}^d \alpha_i = 1, x = y - \lambda \sum_{i=1}^d \alpha_i (z_i - y)\}$
3. Set $\mathcal{B}'_{\tau+1}$ to be the minimum volume ellipsoid containing $\mathcal{B}_\tau \setminus \mathcal{K}_\tau$
4. Set $\mathcal{X}_{\tau+1} = \mathcal{X}_\tau \cap \mathcal{B}'_{\tau+1}$

Output: Output: new feasible region $\mathcal{X}'_{\tau+1}$ and enclosing ellipsoid $\mathcal{B}'_{\tau+1}$

Algorithm HAT-RAISING

Input: pyramid Π with apex y, (rounded) feasible region $\mathcal{X}_\tau$ for each epoch τ, enclosing ball $\mathcal{B}_\tau$

1. Let CENT be the center of Π
2. Set $y' = y + (y - \text{CENT})$
3. Set Π' to be the pyramid with apex y' and same base as Π
4. Set $(\mathcal{X}_{\tau+1}, \mathcal{B}'_{\tau+1}) = \text{CONE-CUTTING}(\Pi', \mathcal{X}_\tau, \mathcal{B}_\tau)$

Output: new feasible region $\mathcal{X}'_{\tau+1}$ and enclosing ellipsoid $\mathcal{B}'_{\tau+1}$

is that we must build estimates of the $CVaR$ of the random loss at every point instead of building them for the expected loss. Because of this, we have to use different concentration results which directly affect how many times we must choose an action.

In this section we will first prove the correctness of the algorithm and then bound the regret. As in the 1-dimensional case we work conditioned on $\mathcal{E}$ which is defined as the event that for every epoch and every round i, $h(x) \in [LB_{\gamma_i}(x), UB_{\gamma_i}(x)]$ for all x played in that round. We will assume that

$$\Delta_\tau(\gamma) = \big(\frac{6c_1 d^4}{c_2^2} + 3\big)\gamma \text{ and } \bar{\Delta}_\tau(\gamma) = \big(\frac{6c_1 d^4}{c_2^2} + 5\big)\gamma \tag{2.10}$$

and $c_1 \geq 64$, $c_2 \leq 1/32$.

Correctness of the Algorithm

In the next sequence of lemmas we show that whenever the cone-cutting procedure is carried out we do not discard all the approximate optima of h. We also show that the hat-

raising step does what we claim.

For the next two lemmas we assume that the distance from apex y of any Π built in epoch τ to the center of $\mathbb{B}(r_\tau)$ is at least r_τ/d. That the previous is true will be shown later.

Lemma 13. *Let $\mathcal{K}_\tau$ be the cone that will be discarded in epoch τ through case 1b) in round i. Let* BOTTOM *be the lowest CI of pyramid Π. Assume the distance from the apex y to the center of $\mathbb{B}(r_\tau)$ is at least r_τ/d. Then $h(x) \geq h(\text{BOTTOM}) + \gamma_i \; \forall x \in \mathcal{K}_\tau$.*

Proof. Let x be a point in $\mathcal{K}_\tau$. By construction, there exists a point z in the base of the pyramid such that $x = \alpha z + (1-\alpha)y$ for some $\alpha \in (0, 1]$. Using the convexity of h, the fact that z is in the base, and the fact that we are in case 1b), we have the two following inequalities

$$h(z) \leq h(\text{TOP}) \leq h(y) + 3\hat{\gamma}$$
$$h(y) \geq h(\text{BOTTOM}) + \Delta_\tau(\hat{\gamma}) - 2\hat{\gamma}$$

where $\hat{\gamma}$ is the CI level used for the pyramid. Since h is convex we have

$$h(y) \leq \alpha h(z) + (1-\alpha)h(x) \leq \alpha(h(y) + 3\hat{\gamma}) + (1-\alpha)h(x).$$

Which implies

$$h(x) \geq h(y) - 3\frac{\alpha}{1-\alpha}\hat{\gamma} > h(\text{BOTTOM}) + \Delta_\tau(\hat{\gamma}) - 3\frac{\alpha}{1-\alpha}\hat{\gamma} - 2\hat{\gamma}.$$

We know $\alpha/(1-\alpha) = ||y-x||/||y-z||$. Since $x \in \mathbb{B}(R_\tau)$, $||y-x|| \leq 2R_\tau = 2c_1 d r_\tau$. Moreover, $||y-z||$ is at least the height of Π, which by Lemma 73 in the Appendix, is at least $r_\tau c_2^2/d^3$. Thus

$$\frac{\alpha}{1-\alpha} = \frac{||y-x||}{||y-z||} \leq \frac{2c_1 d r_\tau}{r_\tau c_2^2/d^3}.$$

This implies

$$h(x) > h(\text{BOTTOM}) + \Delta_\tau(\hat{\gamma}) - 2\hat{\gamma} - \frac{6c_1 d^4}{c_2^2}\hat{\gamma} \geq h(\text{BOTTOM}) + \gamma_i$$

as required. □

Lemma 14. *Let Π' be the pyramid built using the hat-raising procedure with apex y' and the same base as Π in round i of epoch τ. let $\mathcal{K}'_\tau$ be the cone to be removed. Assume the distance from y, the apex of Π to the center of $\mathbb{B}(r_\tau)$ is at least r_τ/d. Then Π' has angle $\bar{\phi}$ at the apex with $\cos\bar{\phi} \leq 2c_2/d$, height at most $2r_\tau c_1^2/d^2$, and every point x in $\mathcal{K}'_\tau$ satisfies $h(x) \geq h(x^*) + \gamma_i$.*

Proof. Let $y' = y + (y - \text{CENTER})$ be the apex of Π'. Let g be the height of Π (the shortest distance from the apex to the base), let g' be the height of Π' and let b be the distance from any vertex in the base to the center of the base. By Lemma 73 in the Appendix we have $g' < 2g \leq 2r_\tau c_1^2/d^2$. Since $\cos\phi = g/\sqrt{h^2+b^2} = c_2/d$ we have $\cos\bar{\phi} = g'/\sqrt{g'^2+b^2} \leq 2g/\sqrt{g^2+b^2} = 2\cos\phi = 2c_2/d$.

We now show that for all $x \in \mathcal{K}'_\tau$ we have $h(x) \geq h(x^*) + \hat{\gamma}$. Since h is convex we have $h(y) \leq (h(y) + h(\text{CENTER}))/2$ therefore $h(y') \geq 2h(y) - h(\text{CENTER})$. Since we are in case 2b) we know $h(\text{CENTER}) \leq h(y) - \bar{\Delta}_\tau(\hat{\gamma})$, so

$$h(y') \geq h(y) + \bar{\Delta}_\tau(\hat{\gamma}). \tag{2.11}$$

Since we are under case 2b) we have $h(y) > h(\text{TOP}) - \Delta_\tau(\hat{\gamma}) - 2\hat{\gamma} > h(x) - \Delta_\tau(\hat{\gamma}) - 2\hat{\gamma}$ for all $x \in \Pi$. We therefore have that for any z in the base of Π,

$$h(y') > h(z) + \bar{\Delta}_\tau(\hat{\gamma}) - \Delta_\tau(\hat{\gamma}) - 2\hat{\gamma} \geq h(z), \tag{2.12}$$

where we used the settings of $\Delta_\tau(\hat{\gamma})$ and $\bar{\Delta}_\tau(\hat{\gamma})$. Finally, for any $x \in \mathcal{K}'_\tau$ there exists $\alpha \in [0,1)$ and z in the base of Π' such that $y' = \alpha z + (1-\alpha)x$, by convexity we have

$h(y') \leq \alpha h(z) + (1-\alpha)h(x) \leq \alpha h(y') + (1-\alpha)h(x)$. The previous implies $h(x) \geq h(y') \geq h(y) + \bar{\Delta}_\tau(\hat{\gamma}) \geq h(x^*) + \gamma_i$. □

Regret Analysis

As in the 1-dimensional case, to bound the total pseudo-regret ($T\bar{\mathcal{R}}_T$) we must bound the regret incurred in a round and then bound the total number of epochs. In this section, for ease of reading we refer to quantity $T\bar{\mathcal{R}}_T$ as the regret.

Bounding the regret incurred in a round.

We first bound the regret in round i if case 2a) takes place. As before, we let Π be a pyramid built by the algorithm with angle ϕ, apex y, base $z_1, ..., z_d$ and center CENTER. recall that the pyramids built by the algorithm are such that the distance from the center to the base is at least $r_\tau c_2^2/d^3$.

Lemma 15. *Suppose the algorithm reaches case 2a) in round i of epoch τ, assume $x^* \in \mathcal{B}(R_\tau)$, where x^* minimizes h. Let Π be the current pyramid and $\hat{\gamma}$ be the current width of the CI. Assume the distance from the apex of Π to the center of $\mathcal{B}(r_\tau)$ is at least r_τ/d. Then the regret incurred while playing on Π in round i is no more than*

$$\frac{\kappa d \ln(T/(\alpha\hat{\gamma}))}{\alpha^2\hat{\gamma}}\Big(\frac{4d^7c_1}{c_2^3} + \frac{d(d+2)}{c_2}\Big)\Big(\frac{12c_1d^4}{c_2^2} + 11\Big).$$

Proof. The proof follows by convexity. We will first bound the variation of h in the pyramid and then bound the regret on the round depending on wether x^* is in Π or not.

Since Π is a convex set we know that the function value on any point in Π is bounded above by the maximum function value at the vertices. Case 2a) implies that for any vertex its function value is bounded above by $h(\text{CENTER}) + \Delta_\tau(\hat{\gamma}) + \bar{\Delta}_\tau(\hat{\gamma}) + 3\hat{\gamma}$. The previous

implies that for all $x \in \Pi$ we have

$$h(x) \leq h(\text{CENTER}) + \Delta_\tau(\hat{\gamma}) + \hat{\Delta}_\tau(\hat{\gamma}) + 3\hat{\gamma}.$$

We let $\delta := \Delta_\tau(\hat{\gamma}) + \hat{\Delta}_\tau(\hat{\gamma}) + 3\hat{\gamma}$. Let $x \in \Pi$, let b be the a point in the base of Π such that $\text{CENTER} = \alpha x + (1-\alpha)b$ for some $\alpha \in [0,1]$. We know that $(1-\alpha)/\alpha = ||\text{CENTER} - x||/||\text{CENTER} - b||$. Since the furthest x can be from CENTER is when x is a vertex, and the distance from CENTER to b is at least the radius of the largest ball inscribed in Π with center CENTER, by Lemma 74 in the Appendix we have

$$\frac{1-\alpha}{\alpha} = \frac{||\text{CENTER} - x||}{||\text{CENTER} - b||} \leq \frac{d(d+1)}{c_2}$$

Since h is convex and we have a bound on all the function values over Π we have

$$h(\text{CENTER}) \leq \alpha h(x) + (1-\alpha)h(b) \leq \alpha h(x) + (1-\alpha)(h(\text{CENTER}) + \delta).$$

This implies

$$h(x) \geq h(\text{CENTER}) - \frac{d(d+1)\delta}{c_2}. \tag{2.13}$$

Combining the previous two equations we have that for any $x, x' \in \Pi$

$$|h(x) - h(x')| \leq \frac{d(d+2)\delta}{c_2}.$$

Consider the case when $x^* \in \Pi$. Since in a given round we sample $d+2$ points in the pyramid, each of them only $\kappa \ln(T/(\alpha\hat{\gamma}))/(\alpha^2\hat{\gamma}^2))$ we have that the total regret incurred when sampling the pyramid is no more than

$$(d+2)(\frac{d(d+2)\delta}{c_2})(\frac{\kappa \ln(T/(\alpha\hat{\gamma}))}{\alpha^2\hat{\gamma}^2}).$$

We now consider the case where $x^* \notin \Pi$. Recall that we always have $x^* \in \mathcal{B}_\tau$ by Lemma 13. Thus we can write $b = \alpha x^* + (1-\alpha)\text{CENTER}$, for some $\alpha \in [0,1]$ where b is a point in some face of the current pyramid. We know $\alpha = ||\text{CENTER} - b||/||\text{CENTER} - x^*||$. Using the triangle inequality we have $||\text{CENTER} - x^*|| \leq 2R_\tau = 2c_1 d r_\tau$. We also know that $||\text{CENTER} - b||$ is at least the radius of the largest ball inscribed in Π which by 74 in the Appendix is at least $r_\tau c_2^2/(2d^4)$. Using the convexity of h and Equation (2.13) we have

$$h(\text{CENTER}) - \frac{d(d+2)\delta}{c_2} \leq h(b) \leq \alpha h(x^*) + (1-\alpha)h(\text{CENTER}).$$

Thus, $\forall x \in \Pi$ we have

$$h(x^*) \geq h(\text{CENTER}) - \frac{d(d+1)\delta}{c_2\alpha} \geq h(\text{CENTER}) - \frac{4d^7 c_1 \delta}{c_2^3} \geq h(x) - \frac{4d^7 c_1 \delta}{c_2^3} - \frac{d(d+2)\delta}{c_2}.$$

Using the same argument as before we know that the regret incurred in the round while evaluating points in Π is no more than

$$(d+2)(\frac{4d^7 c_1 \delta}{c_2^3} + \frac{d(d+2)\delta}{c_2})(\frac{\kappa \ln(T/(\alpha\hat{\gamma}))}{\alpha^2 \hat{\gamma}^2}).$$

Plugging in $\Delta_\tau(\hat{\gamma})$ and $\bar{\Delta}_\tau(\hat{\gamma})$ yields the result. □

Lemma 15 is important because it implies that whenever we sample from a pyramid using $\hat{\gamma}$ we were in Case 2a) with $2\hat{\gamma}$ and the regret incurred is only $poly(d)/\hat{\gamma}$. The exception is when we are in the first round, however since h is 1-Lipschitz the previous claim holds trivially.

We now show that we only visit Case 1a) only a bounded number of times in every round. The intuition is that every time Case 1a) occurs and we build a new pyramid its center will be closer to the center of $\mathcal{B}(R_\tau)$ and at some point the pyramid will be inside the simplex we built at the beginning of the epoch for which we know h at its vertices.

Lemma 16. *At any round, the number of visits to Case 1a) is at most* $2d^2\ln(d)/c_2^2$*, and every pyramid build by the algorithm with apex* y *satisfies* $||y - x_0|| \geq r_\tau/d$*.*

Proof. By definition of Case 1a) TOP $\neq y$, without loss of generality we assume TOP $= z_1$. By construction we have

$$||z_1 - x_0|| = \sin(\phi)||y - x_0||.$$

Since this holds every time we enter Case 1a), we know that the total number of visits k satisfies

$$||z_1 - x_0|| = (\sin(\phi))^k r_\tau$$

where r_τ is the radius of the ball where the simplex is inscribed at the beginning of round τ. We also notice that for a simplex of radius r_τ the largest ball inscribed in it has radius r_τ/d. Additionally, by construction we have $\cos(\phi) = c_2/d$ and therefore $\sin(\phi) = \sqrt{1 - c_2^2/d} \leq 1 - c_2^2/(2d^2)$. Therefore, $k = 2d^2\ln(d)/c_2^2$ ensures $||z_1 - x_0|| \leq r_\tau/d$ which implies that z_1 lies inside the simplex we build at the beginning of round τ.

Let $y_1, ..., y_k$ be the apexes of the pyramids built in round τ. By construction we have

$$h(z_1) \geq h(\text{TOP}) \geq h(y_k)\gamma \geq h(y_{k-2})2\gamma \geq ... \geq h(y_1) + k\gamma.$$

On the other hand, by definition of y_1 we have $h(y_1) \geq h(x_i) - \gamma$ for all vertices of the simplex x_i. Since z_1 is in the simplex and h is convex we have

$$h(y_1) \geq h(z_1) - \gamma \geq h(y_1) + (k-1)\gamma$$

which is a contradiction unless $k \leq 1$. Therefore, if z_1 is not in the simplex it must be the case that $k \leq 2d^2\ln(d)/c_2^2$. □

Using the Lemma 16 we will bound the regret incurred in a round whenever it terminates in Case 2a).

Lemma 17. *For any round with CI width of γ that terminates in Case 2a) the total regret incurred in the round is no more than*

$$\frac{\kappa d \ln(T/(\alpha\gamma))}{\alpha^2\gamma}\big(\frac{2d^2\ln(d)}{c_2^2}+1\big)\big(\frac{4d^7c_1}{c_2^3}+\frac{d(d+2)}{c_2}\big)\big(\frac{12c_1d^4}{c_2^2}+11\big).$$

Proof. By Lemma 16 we have that for the given round, the total number of pyramids we have built is $k \le 2d^2\ln(d)/c_2$. Then, by Lemma 15 we know that for any point in the k-th pyramid the instantaneous regret is no more than

$$\delta := \kappa\gamma d\big(\frac{4d^7c_1}{c_2^3}+\frac{d(d+2)}{c_2}\big)\big(\frac{12c_1d^4}{c_2^2}+11\big).$$

We now show that the regret for any point we played during the round is at most δ. Indeed, by construction y_k is TOP of the $(k-1)$-th pyramid. By definition of Case 1a) we know that for any $x \in \Pi_{k-1}$ we have $f(x) \le f(y_k)+\gamma$. Using this reasoning, we get that the function value at any vertex of any pyramid we have built during the round is also bounded by the function value at y_k. Additionally, as in the proof of the previous lemma, the function value at all the vertices of the simplex we built at the beginning of the epoch is also bounded by the function value at y_k. Since in every pyramid (and the initial simplex) we sample $d+2$ points we know that the total number of points we will play at is no more than $(d+2)(2d^2/(c_2^2\ln(d))+1)$. To bound the total number of times we play a point we notice that for a CI with width $\hat{\gamma}$ we play it $\kappa\ln(T/(\alpha\gamma))/(\alpha^2\hat{\gamma}^2)$. Suppose $\gamma = 2^{-i}$, since $\hat{\gamma}$ is geometrically decreased to γ we know that the total number of plays at any point is bounded by

$$\sum_{j=1}^{i}\frac{\kappa\ln(T/(\alpha\gamma))}{\alpha^2 2^{-2j}} \le \frac{4\kappa\ln(T/(\alpha\gamma))2^{2i}}{\alpha^2} = \frac{4\kappa\ln(T/(\alpha\gamma))}{\alpha^2\gamma^2}$$

Putting everything together we get that the total regret incurred during the round is no more than

$$\frac{\kappa d \ln(T/(\alpha\gamma))}{\alpha^2\gamma}\big(\frac{2d^2\ln(d)}{c_2^2}+1\big)\big(\frac{4d^7c_1}{c_2^3}+\frac{d(d+2)}{c_2}\big)\big(\frac{12c_1d^4}{c_2^2}+11\big).$$

□

Using Lemma 17 we will now bound the total regret incurred at any round.

Lemma 18. *For any round that terminates in a CI with width γ, the total regret over the round is no more than*

$$\frac{\kappa d \ln(T/(\alpha\gamma))}{\alpha^2\gamma}\big(\frac{2d^2\ln(d)}{c_2^2}+1\big)\big(\frac{4d^7c_1}{c_2^3}+\frac{d(d+2)}{c_2}\big)\big(\frac{12c_1d^4}{c_2^2}+11\big).$$

Proof. We just need to bound the regret when the round ends in Case 1b) or 2b). By the definition of the algorithm, whenever a round has level γ it must be the case that in the previous round the level was 2γ and thus using the previous lemma we can bound the regret. The exception is in the first round when $\gamma = 1$, in this case using the Lipschitz assumption we know that the instantaneous regret is no more than 1.

Because of the previous we have that the instantaneous regret at any point of the simplex we build is no more than

$$2\gamma\big(\frac{4d^7c_1}{c_2^3}+\frac{d(d+2)}{c_2}\big)\big(\frac{12c_1d^4}{c_2^2}+11\big).$$

Now, if the algorithm was in Cases 1a), 1b) , or 2b) with level $\hat{\gamma}$, then it must have been in Case 2a) with level $2\hat{\gamma}$. And thus, using the bound on the regret whenever a round ends through Case 2a), we have that the instantaneous regret on the vertices any pyramid is no more than

$$2\hat{\gamma}\big(\frac{4d^7c_1}{c_2^3}+\frac{d(d+2)}{c_2}\big)\big(\frac{12c_1d^4}{c_2^2}+11\big),$$

and by using the same argument as in the proof of Lemma 17, the number of plays at a given point is bounded above by $\kappa \ln(T/(\alpha\gamma))/(\alpha^2\hat{\gamma}^2)$. Therefore, the total regret incurred at any pyramid built by the algorithm is no more than

$$\frac{\kappa d \ln(T/(\alpha\hat{\gamma}))}{\alpha^2\gamma}\big(\frac{4d^7c_1}{c_2^3} + \frac{d(d+2)}{c_2}\big)\big(\frac{12c_1d^4}{c_2^2} + 11\big).$$

Recalling the bound on the total number of pyramids built in any round yields the result.

□

Lemma 19. *The regret in any epoch which ends in level γ is at most*

$$\frac{\kappa d \ln(T/(\alpha\gamma))}{\alpha^2\gamma}\big(\frac{2d^2\ln(d)}{c_2^2} + 1\big)\big(\frac{4d^7c_1}{c_2^3} + \frac{d(d+2)}{c_2}\big)\big(\frac{12c_1d^4}{c_2^2} + 11\big).$$

Proof. From Lemma 18 we know that on any round with level γ, the regret is bounded by C/γ where C is some constant. Since γ is reduced geometrically, the net regret on an epoch where the largest level we encounter is γ is bounded by

$$\sum_{j=1}^{i} \frac{C}{2^{-j}} \leq 2C2^i = \frac{2C}{\gamma},$$

which yields the result. □

Bounding the Number of Epochs

To bound the number of epochs we must show that every time CONE-CUTTING is performed we discard a sufficiently large portion of the current ball. More specifically, we need to analyze the ratios of volumes of $\mathcal{B}_{\tau+1}$ and $\mathcal{B}_\tau$.

Lemma 20. *Let $\mathcal{B}_\tau$ be the smallest ball containing $\mathcal{X}_\tau$, let $\mathcal{B}'_{\tau+1}$ be the minimum volume ellipsoid containing $\mathcal{B}\tau \setminus \mathcal{K}_\tau$. Then, for small enough constants c_1, c_2, $vol(\mathcal{B}'_{\tau+1}) \leq \rho \cdot vol(\mathcal{B}_\tau)$ where $\rho = \exp(-\frac{1}{4(d+1)})$.*

Proof. This result is analogous to the volume reduction results for the ellipsoid method with a gradient oracle. It is easy to see that it suffices to consider the intersection of $\mathcal{B}_\tau$ with a half-space in order to understand the set $\mathcal{B}_\tau \setminus \mathcal{K}_\tau$. This is because if we were to discard only the spherical cap instead of the whole cone then the minimum enclosing ellipsoid would increase its volume.

The previous choices of c_1, c_2 guarantee that the distance from the center of $\mathcal{B}_\tau$ to the origin is at most $R_\tau/(4(d+1))$. The previous is true because by construction the apex of cone $\mathcal{K}_\tau$ is always contained in $\mathbb{B}(r_\tau)$, and the height of the cone is at most $R_\tau \cos(\bar{\phi}) \leq R_\tau/(8(d+1))$ again by construction. Thus, if $r_\tau \leq R_\tau/(32(d+1))$, then the distance of the hyperplane to the origin is at most $R_\tau/(4(d+1))$.

Therefore, $\mathcal{B}'_{\tau+1}$ is the minimum volume ellipsoid that contains the intersection of $\mathcal{B}_\tau$ with a hyperplane that is at most $R_\tau/(4(d+1))$ from its center. Using Theorem 2.1 from [61] (with $\alpha = -1/(4(d+1))$) we get the result. □

Lemma 21. *At any epoch with CI level γ, the instantaneous regret of any point in $\mathcal{K}_\tau$ is at least γ.*

Proof. Since every epoch terminates only through Cases 1b) or 2b) we only check the claim is true for these two cases. If the epoch ends through Case 1b) the proof of Lemma 13 gives the result. If the epoch ends through Case 2b), after HAT-RAISING we now that the apex y' of pyramid Π' satisfies $h(y') \geq h(z_i) + \gamma$ for all vertices $z_1, ..., z_d$ of the pyramid. Writing $y' = \alpha x + (1-\alpha)z$ with x in $\mathcal{K}_\tau$, z in the base of Π' and $\alpha \in [0,1]$, we can conclude that $h(x) \geq h(x^*) + \gamma$ just as we did in the proof of Lemma 14. □

We are now ready to bound the total number of epochs.

Lemma 22. *The total number of epochs in the algorithm is no more than $\frac{d\ln(T)}{\ln(1/\theta)}$ where $\theta = \exp(-\frac{1}{4(d+1)})$.*

Proof. Recall x^* is the minimizer of h. Since h is 1-Lipschitz, any point inside a ball or radius $1/\sqrt{T}$ centered around x^* has instantaneous regret of at most $1/\sqrt{T}$. The volume of

this ball is $T^{-d/2}V_d$, with V_d equal to the volume of the unit ball in d-dimensions. Suppose the algorithm goes through k epochs. By Lemma 20 we know that the volume of $\mathcal{X}_\tau$ is bounded above by $\rho^k V_d$. By the previous lemma we know that the instantaneous regret of any point that was discarded had instantaneous regret at least $1/\sqrt{T}$. This is because at any given epoch and round we sample at $\frac{\kappa \ln(T/(\alpha\gamma))}{\alpha^2\gamma^2}$ and this quantity can not be more than T. Because of the previous, any point in the ball centered at x^* with radius $1/\sqrt{T}$ is never discarded. Therefore the algorithms stops whenever

$$\theta^k V_d \leq T^{-d/2} V_d$$

implying $k \leq \frac{d\ln(T)}{\ln(1/\theta)}$. □

We are now ready to prove Theorems 6 and 7.

Proof of Theorem 6. Using the bound on the regret incurred in an epoch and the fact that $\gamma \geq 1/\sqrt{T}$ we know the total regret on an epoch is no more than

$$\frac{\kappa d\sqrt{T}\ln(T/\alpha)}{\alpha^2}\big(\frac{2d^2\ln(d)}{c_2^2}+1\big)\big(\frac{4d^7c_1}{c_2^3}+\frac{d(d+2)}{c_2}\big)\big(\frac{12c_1d^4}{c_2^2}+11\big).$$

By the previous lemma we know the total number of epochs is no more than $d\ln(T)/\ln(1/\theta)$. Thus the total regret $T\bar{\mathcal{R}}_T$ is bounded above by

$$\frac{\kappa d^2\sqrt{T}\ln(T/\alpha)\ln(T)}{\alpha^2\ln(1/\theta)}\big(\frac{2d^2\ln(d)}{c_2^2}+1\big)\big(\frac{4d^7c_1}{c_2^3}+\frac{d(d+2)}{c_2}\big)\big(\frac{12c_1d^4}{c_2^2}+11\big).$$

Recall that we were working conditioned on $\mathcal{E}$. As in the proof of the 1-dimensional algorithm, we have $P(\mathcal{E}') \leq 1/T$. Plugging in the value of θ above yields the result. □

Proof of Theorem 7. The proof is almost the same as the one of Theorem 5 with two slight differences. First, we use Theorem 6, instead of 4 to bound $\bar{\mathcal{R}}_T$. Second, using the same argument as in the proof of Theorem 5 we get that with probability at least $1-\frac{2}{T}$, $CE =$

$\tilde{O}(\frac{\sqrt{d}}{\alpha\sqrt{T}})$. $\square$

2.8 Extension to More General Risk Measures

In Sections 2.6 and 2.7 we developed regret minimization algorithms suitable for decision makers who are risk averse, where the notion of risk was measured using the $CVaR_\alpha$. In this section we extend our results to more general risk measures. We slightly modify the setup from Section 2.4. Now, we assume ξ is a discrete random variable supported in Ξ with $|\Xi| = N$. That is, there are N scenarios. Moreover we assume that each scenario has the same probability of occurring. Let $X \subset \mathbb{R}^d$ be a convex and compact set. Let $f : X \times \Xi \to \mathbb{R}$ be a convex function in the first argument for every $\xi \in \Xi$. Let f satisfy $||\nabla f(x, \xi)|| \leq G$ for every $\xi \in \Xi$ and every $x \in X$. Additionally, we assume $0 \leq f(x, \xi) \leq 1$ for every $x \in X$ and every $\xi \in \Xi$. We consider some law invariant, coherent and comonotone risk measure $\rho(\cdot)$. Our goal now is to obtain vanishing pseudo-ρ-regret

$$\bar{\mathcal{R}}_T^\rho := \frac{1}{T}\sum_{t=1}^{T} \rho[F](x_t) - \frac{1}{T}\min_{x\in X}\sum_{t=1}^{T}\rho[F](x),$$

and ρ-regret

$$\mathcal{R}_T^\rho := \rho[\{f_t(x_t)\}_{t=1}^T] - \min_{x\in X}\rho[\{f_t(x_t)\}_{t=1}^T].$$

In this section we will show that by using the Kusuoka Representation Theorem along with the ideas we developed earlier we can obtain vanishing $\bar{\mathcal{R}}_T^\rho$ and $\mathcal{R}_T^\rho$.

2.8.1 Kusuoka Representation of Risk Measures

In this section we consider some risk measures and present a classical result from Kusuoka [89].

Definition 1. *A risk measure $\rho : \mathcal{X}(\Omega, 2^{\Omega}, P) \to \mathbb{R}$ is coherent if for every $X_1, X_2 \in \mathcal{X}$ it is:*

- *Normalized,* $\rho(0) = 0$.
- *Monotone,* $X_1 \leq X_2 \implies \rho(X_1) \leq \rho(X_2)$.
- *Superadditive,* $\rho(X_1) + \rho(X_2) \leq \rho(X_1 + X_2)$.
- *Positive homogenous,* $\rho(\lambda X_1) = \lambda \rho(X_1), \forall \lambda > 0$.
- *Translation invariant,* $\rho(X_1 + c) = \rho(X_1) + c$.

Moreover, we say ρ is *law invariant* if $\rho(X_1)$ depends only on the distribution of X_1. Additionally, we say ρ is *comonotone additive* if $\rho(X_1 + X_2) = \rho(X_1) + \rho(X_2)$.

It is well known [7] that $CVaR$ is a coherent risk measure. Indeed many risk measures can be expressed as functions of $CVaR$ Pichler and Shapiro [108]. We present a special case of the Kusuoka representation theorem that will be useful later.

Lemma 23. *[107] Consider a finite probability space $(\Omega, 2^{\Omega}, P)$, with $\Omega = \{\omega_1, ..., \omega_N\}$, and $P(\omega_n) = \frac{1}{N}$. Assume P is such that $P(\omega_n) = \frac{1}{N}$, $\forall n = 1, ..., N$. Then, a mapping $\rho : \mathcal{X}(\Omega, 2^{\Omega}, P) \to \mathbb{R}$ is a law invariant coherent and comonotone additive risk measure if and only if it has a Kusuoka representation of the form*

$$\rho(X) = \sum_{n=1}^{N} \mu_n CVaR_{\frac{n}{N}}(X), \quad \forall X \in \mathcal{X} \tag{2.14}$$

where $\mu \in [0, 1]^N$ and $||\mu||_1 = 1$.

Pichler and Shapiro [108] give examples on how the Kusuoka representation theorem can be used, in particular how to write the following risk measures as mixtures of $CVaR$'s. We refer the reader to their paper for the details.

- $\rho(Z) := \inf_{t \in \mathbb{R}} \{t + c||[Z - t]_+||_p\}, \quad \forall Z \in \mathcal{L}^p(\omega, \mathcal{F}, P)$ with $c > 1$ and $1 < p < \infty$.
- $\rho(Z) := \mathbb{E}[Z] + \lambda||[Z - \mathbb{E}[Z]]_+||$ for $p \geq 1$ and $0 \leq \lambda \leq 1$.

2.8.2 Algorithms

We define for every $t = 1, ..., T$, function $\mathcal{G}_t(x, z) : X \times Z \to \mathbb{R}$, with $Z := [0, 1]^N$, as

$$\mathcal{G}_t(x, z) := \sum_{n=1}^{N} \mu_n (z_n + \frac{1}{n/N}[f_t(x) - z_n]_+)$$

for some $\mu \in [0, 1]^N, \mu \geq 0, ||\mu||_1 = 1$. For convenience we write $\mathcal{L}_n^t(x, z) := z_n + \frac{1}{n/N}[f_t(x) - z_n]_+$ for $n = 1, ..., N$. Notice that for any $x \in X$, after taking expectation with respect to ξ and plugging the minimizer of every individual term $\mathcal{L}_n^t$ we end up with the Kusuoka representation of a law invariant, coherent and commonotone risk measure. Let μ be the vector corresponding to the Kusuoka representation of our risk measure of interest ρ (see Equation (2.14)). Algorithm 3, a generalization of Algorithm 1 that uses functions $\mathcal{G}_t$ instead of $\mathcal{L}_t$ can be found in the appendix. We have the following guarantees for Algorithm 3.

Theorem 8. *Algorithm 3 with $\eta = O(\frac{1}{dN^{3/2}T^{3/4}})$ and $\delta = O(\frac{N^{1/2}}{T^{1/4}})$guarantees*

$$\mathbb{E}[\bar{\mathcal{R}}_T^\rho] \leq O(\frac{dN^{3/2}}{T^{1/4}}),$$

where the expectation is taken over the random draw of functions and the internal randomization of the algorithm.

Theorem 9. *Let $f(\cdot, \xi)$ be strongly convex with parameter $\beta > 0$ for all $\xi \in \Xi$. Algorithm 3, run with the same parameters as in Theorem 8, guarantees*

$$\mathbb{E}[\mathcal{R}_T^\rho] \leq O(\frac{d^{1/2}N^{7/4}}{\beta^{1/2}T^{1/8}}),$$

where the expectation is taken over the random draw of functions and the internal randomization of the algorithm.

To obtain a better dependence on the number of rounds T, Algorithm 2 (in both cases,

$d = 1$ and $d > 1$) can be modified to solve this more general problem. The only modification is that we will sample $\tilde{O}(\frac{N^2 \ln(\sqrt{N}T)}{\gamma})$ times a point to build a γ-CI for $\rho[F](x)$ for any $x \in X$. Let this modification of Algorithm 2 be Algorithm 4. We have the following guarantees.

Theorem 10. *Algorithm 4 run with the right parameters guarantees that with probability at least* $1 - \frac{1}{T}$

$$\bar{\mathcal{R}}_T^{\rho} \leq \tilde{O}(\frac{N^2 poly(d)}{\sqrt{T}}).$$

Theorem 11. *Let* $f(\cdot, \xi)$ *be strongly convex with parameter* $\beta > 0$ *for all* $\xi \in \Xi$*, Algorithm 4 run with the right parameters guarantees that with probability at least* $1 - \frac{3}{T}$

$$\mathcal{R}_T^{\rho} \leq \tilde{O}(\frac{N^3 poly(d)}{\beta^{1/2} T^{1/4}}).$$

2.9 Experimental Results

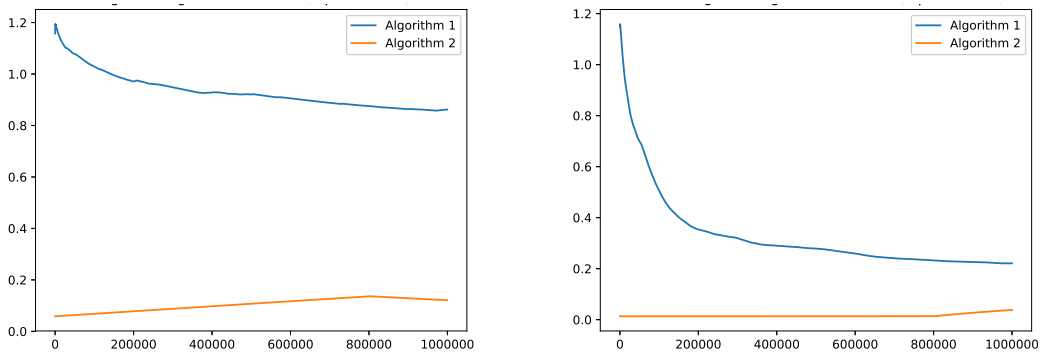

Figure 2.1: Regret (left) and Pseudo Regret (right) of Algorithms 1 and 2 (as a function of T) with $\alpha = 0.01$

In this section we test the performance of Algorithms 1 and 2 and see if they behave as predicted. We first present experimental results for the 1-dimensional case and then for the general d-dimensional case. A more detailed analysis of the experimental results can be found in the appendix.

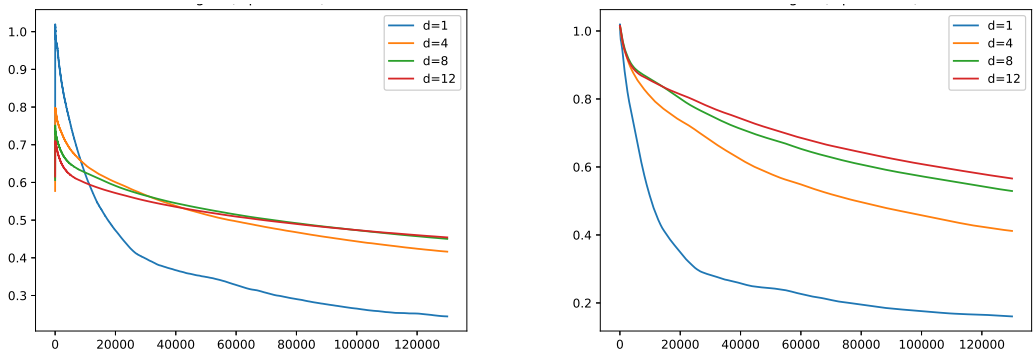

Figure 2.2: Regret (left) and Pseudo Regret (right) of Algorithm 1 (as a function of T) with $\alpha = 0.25$

2.9.1 The 1-Dimensional Case

We tested the algorithms against an instance generated the following way. We let $f(x, \xi) = \frac{1}{x} + (.05 - .04\xi)x^2$ where $\xi \sim U[0, 1]$, that is ξ is sampled uniformly from the interval $[0, 1]$, and $x \in X := [0.5, 6]$. By using Equation (5) some simple algebra yields $C_\alpha[F](x) = \frac{1}{x} + (.05 - .02\alpha)x^2$ with minimum occurring at $x^* = \frac{1}{(.1-.04\alpha)^{1/3}}$. Notice that with the closed form expressions it is easy to evaluate $\bar{\mathcal{R}}_T$. To evaluate $\mathcal{R}_T$ we will approximate the term $\min_{x \in X} C_\alpha[\{f_t(x^*)\}_{t=1}^T]$ with $C_\alpha[\{f_t(x^*)\}]$ which by Lemma 1 (in the appendix) should not be too far. To compute empirical pseudo-regrets and regrets we observe the random losses and iterates generated by the algorithms when they are run for $T = 1,000,000$ rounds. Since the previous quantities may vary every time the algorithm is run, we will run each algorithm 25 times and average the outputs. All the parameters for Algorithm 1 were chosen optimally, and the constants hidden by the O-notation were set to 1. The initial iterate was always set far from x^*, $x_0 = 5.8$ every time the algorithm was run.

From our experimental results we can observe that regret indeed decays at a slower rate than the pseudo regret, as predicted by our theoretical results. As can be seen in Figures 2.1 and 2.2 this behavior becomes more apparent when the value of α is smaller.

2.9.2 The d-Dimensional Case

For $d > 1$ the performance of Algorithm 2 is not satisfactory because the regret of it scales badly with d. Hence we only present results for Algorithm 1. The instance for this section was generated as follows, let $f_i(x_i, \xi_i) = \frac{1}{x_i} + (.05 - .04\xi_i)x_i^2$, then define the loss $f(x, \xi) = \frac{1}{d}\sum_{i=1}^{d} f_i(x_i, \xi_i)$ with $\xi \sim U[0,1]^N$ (i.e. ξ is sampled uniformly from the d-dimensional [0,1] cube). It is easy to see that $C_\alpha[F](x) = \frac{1}{d}\sum_{n=1}^{d} \frac{1}{x_i} + (.05 - .02\alpha)x_i^2$ by the previous subsection. Even though the loss function is a summation of coordinate-wise independent functions the algorithm is not aware of it and thus can not exploit the structure. All the parameters of the algorithm were set optimally except for the constant hidden by the O-notation which was set to 1. The initial iterate was always set far from x^*, $x_0 = [5.8; ...; 5.8]$ every time the algorithm was run. From Figure 2.2 we can observe several things. First, the dimensionality of the problem indeed affects the pseudo-regret and the regret negatively. Second, the smaller the α level the higher the pseudo-regret and regret. Third, the regret seems to vanish at a slower rate than the pseudo-regret. This may indicate that our analysis for the regret may be tight with respect to T.

We notice that when $\alpha \neq 1$, at early rounds the regret seems to increase quickly before it starts dropping. This occurs because at the beginning, even though we are in a region with bad $CVaR_\alpha[F]$, we have not observed many losses and thus the empirical $CVaR$ of the sequence of losses is not necessarily large.

2.10 Conclusions and Open Questions

In this chapter we studied the problem of Online Risk-Averse Optimization with bandit feedback under the assumption that the loss functions were sampled from some distribution and provided two algorithms to solve this problem. As future research directions, it would be interesting to see whether we can drop the strong convexity assumption and still obtain sublinear $CVaR$-regret. It would also be interesting to develop an algorithm such

that its regret does not depend on the number of scenarios when we consider some general risk measure ρ. We leave it as an open question whether it is possible to obtain sublinear $CVaR$-regret when the functions are chosen by an adversary instead of from some unknown distribution.

CHAPTER 3

DIFFERENTIALLY PRIVATE ONLINE SUBMODULAR MINIMIZATION

3.1 Introduction

Every day Machine Learning tools, in particular tools from Online Learning, are being applied to sensitive data from individuals. As such, privacy concerns have arisen. In applications such as clinical trials, online ad placement, personalized pricing, and recommender systems, online learning algorithms are dealing with personal (and possibly highly sensitive) data.

In this chapter, we develop the first algorithms for differentially private online submodular optimization. A function $f : 2^{[n]} \to \mathbb{R}$ mapping from discrete collections of elements to real values is *submodular* if it exhibits the following diminishing returns property: for all sets $S, S' \subseteq [n]$ such that $S' \subseteq S$ and for all elements $i \in [n] \setminus S$,

$$f_t(S' \cup \{i\}) - f_t(S') \geq f_t(S \cup \{i\}) - f_t(S).$$

Submodular functions have several applications in machine learning (see [87] for a survey) and are used extensively used economics because their diminishing returns property captures preferences for substitutable goods and satiation from multiple copies of the same good [17, 126].

In the *Online Submodular Minimization* problem, a sequence of T submodular functions $f_1, \ldots, f_T : 2^{[n]} \to \mathbb{R}$ arrive in an online fashion. At every timestep t, a decision maker chooses a set $S_t \subseteq [n]$ before observing the function f_t. The decision maker then incurs cost $f_t(S_t)$. The decision maker's goal is to minimize her total regret, which is defined

as,

$$\text{Regret}(T) = \sum_{t=1}^{T} f_t(S_t) - \min_{S \subseteq [n]} \sum_{t=1}^{T} f_t(S).$$

That is, her regret is the difference between her total cost across all rounds, and the cost of the best fixed set in hindsight after seeing all the functions. We say that an algorithm for the Online Submodular Minimization problem is *no regret* is the regret (or expected regret for randomized algorithms) is sublinear in T: $\text{Regret}(T) = o(T)$.

We consider two different settings based on the type of informational feedback the decision maker receives in each round. In the *full information setting*, the decision maker observes the entire function f_t after making her choice of S_t. In the *bandit setting*, the decision maker only observes her cost $f_t(S_t)$ and does not receive any additional information about the function f_t. The bandit setting is a more challenging environment because the decision maker has severely restricted information when making decisions, but also captures the reality of many real-world online learning problems where counterfactual outcomes cannot be measured.

We formally incorporate the task of preserving privacy by using the framework of differential privacy. Differential privacy was first defined by [45] for algorithms operating on large static databases, and required that if a single entry in the database changed, then the algorithm would produce approximately the same output. In this work, we view our database as the sequence of submodular functions $f_1, \ldots, f_T$, and the algorithm's output is the sequence of sets $S_1, \ldots, S_T$. We require that if a single function f_t were changed to a different f_t', then the entire sequence of chosen sets would be approximately the same. A formal definition is given in the preliminaries.

The main goal of this chapter is to design differentially private no-regret algorithms for the Online Submodular Minimization problem. There are many applications of online learning problems using sensitive data that could benefit from formal privacy guarantees,

such as clinical drug trials, online ad placement, and personalized pricing. For concreteness, we provide the following motivating example for the study of private online submodular optimization.

Motivating Example. As a concrete motivating example we consider the following online ad placement problem. Online retailers such as Amazon, Walmart, and Target design their websites such that the retailers can offer other products at check out which complement the item the customer is buying. Due to item complimentarities, the utility function of user t, g_t, defined over the possible subset of products the retailer can offer $[n]$, is supermodular. However, displaying too many items may hurt the chance of the user buying something else. At time t, the retailer is choosing S_t that maximizes $f_t(S) := g_t(S) - \sum_{i \in S} p_i$ for each user (where $p_i \in \mathbb{R}$ is the "cost" of displaying a product). The retailers must choose S_t without knowing g_t and they receive only bandit feedback (i.e., they can only observe $g_t(S_t)$, and not $g_t(\cdot)$). The retailer seeks to minimize regret: $\max_{S \in [n]} \sum_{t=1}^{T} f_t(S) - \sum_{t=1}^{T} f_t(S_t)$. Notice that since $\sum_{i \in S} p_i$ is modular, then the retailer has to solve an online submodular minimization problem with bandit feedback. Existing recommender systems have been shown to leak information about users [137], motivating the need for formal privacy guarantees in this settings. Therefore, the retailer will perform this optimization in a differentially private manner to ensure that no information about an individual is leaked to other users.

3.1.1 Main Results

In this chapter we develop the first algorithms for online submodular minimization that preserve differential privacy under full information feedback and bandit feedback that are almost as good as the best non-private algorithms.

We start with the full information setting, where the algorithm can observe the entire function f_t after making its decision at each time t. We give an algorithm in this setting

that is both differentially private and satisfies no regret.

Theorem 12 (Informal)**.** *In the full information setting of Online Submodular Minimization, there is an ϵ-differentially private algorithm that achieves regret:*

$$\mathbb{E}[\text{Regret}(T)] = \tilde{O}\left(\frac{n\sqrt{T}}{\epsilon}\right).$$

This algorithm works by first relaxing each input submodular function to a convex function using the Lovasz extension (defined formally in Section 3.2.1). Our algorithm then simulates a variant of an algorithm for differentially private online convex optimization (due to Smith and Thakurta [121]) run on the sequence of Lovasz extensions. The differential privacy guarantee can be proved almost as it was done in [121]. To prove the regret bound, we show that the relaxation and optimization on convex functions does not increase the regret guarantee by too much. Our algorithm matches the regret bound of [121] for private online convex optimization, and loses only a factor of $\frac{1}{\epsilon}$ relative to the optimal non-private regret bound of [68] for online submodular minimization.

We next consider the bandit setting, which is significantly more challenging and requires a refined analysis. The private online convex optimization algorithm of Smith and Thakurta [121] requires use of the subgradient of the Lovasz extension. However in the bandit setting, the algorithm does not receive enough information to compute the exact Lovasz extension or its subgradients. Instead, we construct an unbiased estimate of the subgradient using the one-point estimation method of [68]. We then apply a variant of the algorithm from [121] to the unbiased estimate of the gradient of the Lovasz extension. This yields a differentially private no-regret algorithm for online submodular minimization in the bandit setting.

Theorem 13 (Informal)**.** *In the bandit setting of Online Submodular Minimization, there is*

an ϵ-differentially private algorithm that achieves regret:

$$\mathbb{E}[\text{Regret}(T)] = \tilde{O}\left(\frac{n^{3/2}T^{2/3}}{\epsilon}\right).$$

The regret guarantee of our algorithm for the bandit setting only loses a factor of $\tilde{O}(\frac{n^{1/2}}{\epsilon})$ relative to the best know non-private regret bound of [68] for online submodular minimization. We actually improve upon the best known regret bound for private online convex optimization [121] which has $O(T^{3/4})$ dependence on T, compared to our $O(T^{2/3})$ guarantee.

3.1.2 Related Work

Our results rely on ideas from [121] and [68]. [121] provides a differentially private algorithm for online convex optimization that achieves a regret rate $\tilde{O}(\frac{\sqrt{nT}}{\epsilon})$ in the full information setting, which is worse than the non-private setting by only a factor of polylog$(T)\sqrt{n}$. Under bandit feedback, they give a modification of their full information algorithm that achieves cumulative regret $\tilde{O}(\frac{nT^{3/4}}{\epsilon})$. One of the key components in our algorithms are modifications of these tools for online convex optimization, which are applied once we have relaxed the submodular functions to their convex Lovasz extensions. [68] provides algorithms for non-private online submodular minimization in both the full information and bandit feedback settings. They design subgradient descent-type algorithms that achieve regret of $O(\sqrt{nT})$ and $O(nT^{2/3})$ in the full information and bandit settings respectively. Our algorithms make use of their one-point gradient estimation technique for the bandit setting. We remark that, to the best of our knowledge, there is no known way to modify subgradient descent-type algorithms, to achieve differential privacy in the online convex bandit problem without damaging the regret bounds by less than polylog(T) factors.

Although our algorithms use these tools, composition of these previous results is not straight-forward. The bound on the variance of the one-point gradient estimator for the Lo-

vasz extension is not the same as that of the estimator used for online convex optimization with bandit feedback, which requires special care in the analysis. If one were to blindly compose the results of [121] and [68], it would yield regret $O(\frac{n^2T^{11/12}}{\epsilon})$ in the bandit setting, instead of the regret rate $O(\frac{n^{3/2}T^{2/3}}{\epsilon})$ that we achieve.

A previous (unpublished) version of the current paper [35] showed that a more careful combination of these tools, which takes into account the variance of the one-point gradient estimator for the Lovasz extension but uses the same analysis as in [121], can only achieve regret $\tilde{O}(\frac{n^{3/2}T^{3/4}}{\epsilon})$ in the bandit setting. This approach was unable to achieve the $\tilde{O}(T^{2/3})$ dependence on T that we achieve here because the analysis of [121] first gave differentially private regret guarantees for strongly convex cost function, and then extended these results to the setting with general convex costs via a regularization trick to ensure strong convexity (See Appendix E.3 from [121]). While this regularization trick allows for low regret, $\tilde{O}(T^{3/4})$, for the problem of private online convex optimization, there were dependencies in the regret bound which make it impossible to obtain the rate of $\tilde{O}(T^{2/3})$ for differentially private online submodular minimization. Our analysis requires additional techniques to achieve this lower regret bound.

Other relevant work includes [78], where the authors design differentially private algorithms for online convex optimization. However, these algorithms only achieve optimal regret rates in some special cases. In [10], the authors provide differentially private algorithms for the special case of online linear optimization with bandit feedback, and obtain regret $\tilde{O}(\frac{\sqrt{T}}{\epsilon})$ which is (almost) optimal. The problem of private online submodular maximization has been studied by [100] and [66]. However, our work cannot be compared to theirs since the problems of minimizing and maximizing a submodular functions are very different. Additionally, these works only consider the offline problem with full information feedback. Finally, [19] studies non-private online submodular maximization only under full information feedback.

3.2 Preliminaries

In this section we present background on submodular functions, and differential privacy that will be useful for our results in later sections.

3.2.1 Submodular Functions

Submodular functions share many properties with both convex and concave functions. They can be thought of as convex functions when one is trying to minimize them, however they also exhibit a diminishing marginal returns property as some concave functions do (i.e., $f(x) = \log x$).

Definition 2 (Submodular function)**.** *A function $f : 2^{[n]} \to [-M, M]$ is* submodular *if for all sets $S, S' \subseteq [n]$ such that $S' \subseteq S$ and for all elements $i \in [n] \setminus S$,*

$$f(S' \cup i) - f(S') \geq f(S \cup i) - f(S).$$

The connection between convex and submodular functions is formalized through the *Lovasz extension* (Definition 4), which extends a submodular function f over $[n]$ to its corresponding convex function $\hat{f}$ over $[0, 1]^n$. The Lovasz extension works by first describing each point in $[0, 1]^n$ as a convex combination of points in $\{0, 1\}^n$, which can be interpreted as subsets of $[n]$. It then defines $\hat{f}(x)$ as the convex combination of f evaluated on the sets associated with x. We first define the necessary notation.

Definition 3 (Maximal chain [68])**.** *A chain of subsets of $[n]$ is a collection of sets $A_0, ..., A_p$ such that $A_0 \subset A_1 \subset \cdots \subset A_p$. A chain is* maximal *if $p = n$. For a maximal chain, $A_0 = \emptyset$, $A_n = [n]$, and there is a unique associated permutation $\pi : [n] \to [n]$ such that $A_{\pi(i)} = A_{\pi(i)-1} \cup \{i\}$ for all $i \in [n]$. For this permutation, we can write $A_{\pi(i)} = \{j \in [n] : \pi(j) \leq \pi(i)\}$ for all $i \in [n]$.*

Define $\mathcal{K} = [0, 1]^n$. For any set $S \subseteq [n]$, let $\mathcal{X}_S \in \{0, 1\}^n$ denote the *characteristic*

vector of S, defined as $\mathcal{X}_S(i) = 1$ if $i \in S$ and 0 otherwise. For any $x \in \mathcal{K}$, there is a unique chain $A_0 \subset \cdots \subset A_p$ such that x can be expressed as a convex combination of the characteristic vectors of the A_i. That is, $\exists \mu_1, \ldots, \mu_p > 0$ such that $x = \sum_{i=0}^{p} \mu_i \mathcal{X}_{A_i}$ and $\sum_{i=0}^{p} \mu_i = 1$. Note that if $p < n$ (i.e., the chain is not maximal), the chain can be extended to a maximal chain by setting $\mu_i = 0$ for all i's corresponding the the subsets of $[n]$ that were not present in the original chain. The chain and the weights can be found in $O(n \ln(n))$ time (see, e.g., Chap. 3 of Bach [17]).

We are now ready to define the Lovasz extension $\hat{f}$ of submodular function f.

Definition 4 (Lovasz extension)**.** *Let $f : 2^{[n]} \to [-M, M]$ be submodular. The* Lovasz extension $\hat{f} : \mathcal{K} \to [-M, M]$ *of f is defined as follows. For each $x \in \mathcal{K}$, let $A_0 \subset \cdots \subset A_p$ be the chain associated with x, and let $\mu_1, \ldots, \mu_p$ be the corresponding weights in the convex combination $x = \sum_{i=0}^{p} \mu_i \mathcal{X}_{A_i}$. Then,*

$$\hat{f}(x) := \sum_{i=0}^{p} \mu_i f(A_i) \quad \forall x \in \mathcal{K}.$$

Equivalently, the Lovasz extension can also be defined by sampling τ uniformly at random from the unit interval $[0, 1]$ and considering level set $S_\tau = \{i : x(i) \geq \tau\}$. Then $\hat{f}(x) = \mathbb{E}_\tau[f(S_\tau)]$ for each $x \in \mathcal{K}$.

We now provide some useful properties of the Lovasz extension.

Lemma 24 ([58, 68])**.** *The Lovasz extension $\hat{f}$ of submodular function f is convex. Additionally, for any $x \in \mathcal{K}$, let $\emptyset = B_0 \subset B_1 \subset \cdots \subset B_n$ be any maximal chain associated with x and let $\pi : [n] \to [n]$ be the corresponding permutation. Then a subgradient g of $\hat{f}$ at x is given by: $g(i) = f(B_{\pi(i)}) - f(B_{\pi(i)-1})$ for all $i = 1, \ldots, n$.*

Lemma 25 ([79])**.** *All subgradients g of the Lovasz extension $\hat{f} : \mathcal{K} \to [-M, M]$ of a submodular function are bounded by $\|g\|_2 \leq \|g\|_1 \leq 4M$.*

3.2.2 Tools from Differential Privacy

Let $\mathcal{F}$ be a class of functions. Let $F = \{f_1, ..., f_T\}$ and $F' = \{f'_1, \ldots, f'_T\}$ be sequences of functions where $f_t, f'_t \in \mathcal{F}$, and $f_t, f'_t : \mathcal{R} \to \mathbb{R}$ for all t. We say F and F' are neighboring sequences if $f_t = f'_t$ for all but at most one $t \in [T]$.

Definition 5 (Differential privacy [45])**.** *An algorithm $\mathcal{A} : \mathcal{F}^T \to \mathcal{R}^T$ is* (ϵ, δ)-differentially private *if for all neighboring sequences $F, F' \in \mathcal{F}^T$ and every subset of the output space $\mathcal{S} \subseteq \mathcal{R}^T$,*

$$\Pr[\mathcal{A}(F) \in \mathcal{S}] \leq e^{\epsilon} P[\mathcal{A}(F') \in \mathcal{S}] + \delta.$$

If $\delta = 0$, we say that $\mathcal{A}$ is ϵ-differentially private.

The following theorem states that differential privacy is robust to *post-processing*: computations performed on the output of a differentially private algorithm are still differentially private.

Theorem 14 (Post-processing [45])**.** *Let $\mathcal{A} : \mathcal{D} \to \mathcal{R}$ be (ϵ, δ)-differentially private, and let $f : \mathcal{R} \to \mathcal{R}'$ be an arbitrary randomized function. Then $f \circ \mathcal{A} : \mathcal{D} \to \mathcal{R}'$ is (ϵ, δ)-differentially private.*

Our results require another differentially private algorithm: Tree-based Aggregation Protocol (TBAP). The Tree-Based Aggregation Protocol [38, 46, 121] is a tool for maintaining differentially private partial sums of vectors arriving in an online sequence. At each time t, TBAP outputs a noisy sum of the input vectors up to time t. A full presentation of the algorithm and its properties is given in Appendix B.1.

The following section (Section 3.2.2) discusses Regularized Follow The Leader, an algorithm from [71] for online convex optimization which is used for online learning. Prior work [121] privatized a variant of this algorithm, Follow The Approximate Leader, to give

a differentially private algorithm for online convex optimization that uses TBAP as a subroutine. It takes in a sequence of strongly convex functions and outputs a sequence of points that minimizes regret.

The Cost of Privacy in Online Convex Optimization

Our algorithm uses the following Regularized Follow the Leader (RFTL) of [71] as a subroutine for online convex optimization. This algorithm is known to achieve low regret (Theorem 15).

Algorithm 6 Regularized Follow The Leader: RFTL($\{f_i\}_{i=1}^T, H, X$)

Input: Online sequence of convex cost functions $\{f_1, ..., f_T\}$ strong convexity parameter H, convex compact decision set $X \subseteq \mathbb{R}^n$.
Output: Sequence of actions $x_1, \ldots, x_T \in X$
Initialize $x_1 \leftarrow \arg\min_{x \in X} \frac{H}{2}||x||_2^2$
Output x_1, observe f_1
for t=1, . . . , T-1 **do**
  $x_{t+1} \leftarrow \arg\min_{x \in X} \sum_{\tau=1}^{t} \nabla f_\tau(x_\tau)^\top x + \frac{H}{2}||x||^2$
  Output x_{t+1} and observe f_{t+1}
end for

Theorem 15 ([72] Ch. 5)**.** *Let $\{f_t\}_{t=1}^T$ be any sequence of convex functions. Let $X \subseteq \mathbb{R}^n$ be a convex and compact set. RFTL guarantees that for any $x \in X$,*

$$\text{Regret}(T) \leq \frac{2}{H} \sum_{t=1}^{T} \|\nabla f_t(x_t)\|^2 + \frac{H}{2} \left[\|x\|^2 - \|x_1\|^2\right].$$

We give the following theorem, which quantifies the loss in regret due to adding a differential privacy constraint. A similar theorem was given in [121] for their analysis of a differentially private version of Follow The Approximate Leader, which is a variant of Regularized Follow the Leader. The main ideas in both proofs are similar, but we analyze a different algorithm (RFTL), so a new proof is needed for Theorem 16.

Theorem 16. *Let $\{\hat{x}_t\}_{t=1}^T$ be the non private iterates of RFTL and let $\{x_t\}_{t=1}^T$ be the private iterates i.e. $x_{t+1} = \arg\min_{x \in X} v_t^\top x + \frac{H}{2}||x||^2$ where v_t is the private partial sum computed*

using $TBAP\{\nabla f_t(x_t), L, \epsilon\}$. *It holds that*

$$\mathbb{E}[\sum_{t=1}^{T} f_t(x_t)] \leq \mathbb{E}[\sum_{t=1}^{T} f_t(\hat{x}_t)] + \frac{4nL^2T\ln^{1.5}(T)}{\epsilon H}.$$

Where the expectation is taken with respect to the randomness of TBAP.

Our proof follows a similar structure as that of Lemma 8 of [121]. However, we analyze a different algorithm, so the proof details are different.

Proof. Let $J_t = v_t^\top x + \frac{H}{2}||x||^2$. Let $\xi_t = v_t - \sum_{\tau=1}^{t} \nabla f_t(x_t)$ be the noise added by TBAP to $\sum_{\tau=1}^{t} \nabla f_t(x_t)$. Notice that $x_{t+1} = \arg\min_{x\in\mathcal{K}} J_t(x) + \xi_t^\top x$ and $\hat{x}_t = \arg\min_{x\in\mathcal{K}} J_t(x)$. Since J_t is $\frac{H}{2}$-strongly convex we have that

$$||\hat{x}_{t+1} - x_{t+1}|| \leq \frac{2||\xi_t||}{H}.$$

Since each ξ_t is formed in TBAP by adding at most $\lceil \ln(T) + 1 \rceil$ vectors with norms drawn from a Gamma distribution with scale n and shape $\frac{(\lceil \ln(T)+1 \rceil)G}{\epsilon}$ we can upper bound $\mathbb{E}[||\xi_t||]$ by $\frac{4nG\ln^{1.5}(T)}{\epsilon}$.

Since f_t is L-Lipschitz continuous we have that

$$\begin{aligned}\mathbb{E}[\sum_{t=1}^{T} f_t(x_t)] &\leq \mathbb{E}[\sum_{t=1}^{T} f_t(\hat{x}_t)] + \mathbb{E}[L\sum_{t=1}^{T} \frac{2||\xi_t||}{H}] \\ &\leq \mathbb{E}[\sum_{t=1}^{T} f_t(\hat{x}_t)] + \frac{4nL^2T\ln^{1.5}(T)}{\epsilon H}.\end{aligned}$$

□

3.3 Full Information Setting

In this section we present Submodular Private Regularized Follow The Leader (SUBMODPFTAL) which is an algorithm for Online Submodular Minimization that is both differentially private and achieves near optimal regret. In the full information setting, the

result follows easily from using RFTL together with TBAP to compute a private version of the sum $\sum_{j=1}^{t} \nabla f_j(x_j)$.

The main difference between using a Regularized Follow The Leader type algorithm versus the subgradient descent type algorithm of [68] is the following. When using SUBMODPFTAL to make the decision at time $t+1$, we use all the subgradients we have observed at times $1, \ldots, t$. To contrast, if we used the algorithm of [68], we would only be using the subgradient obtained at t. This difference is crucial when trying to incorporate privacy into the setting.

Algorithm 7 Submodular Private Regularized Follow The Leader: SUBMODPFTAL($\{f_i\}_{i=1}^T, M, H, L, [n], \epsilon$)

Input: Online sequence of submodular cost functions $\{f_1, ..., f_T\}$, lower and upper bounds function values $[-M, M]$, strong convexity parameter H, Lipschitz parameter L, ground set $[n]$, privacy parameter ϵ.
Output: Sequence of sets $S_1, \ldots, S_T \subseteq [n]$
Initialize $S_1 \leftarrow \emptyset$
Set $x_1 \leftarrow 0 \in \mathcal{K}$
Output S_1
Compute and pass $\nabla \hat{f}_1(x_1)$ to TBAP($\{\nabla \hat{f}_i(x_i)\}, L, \epsilon$), and receive current partial sum v_1
for t=1, . . . , T-1 **do**
  $x_{t+1} \leftarrow \arg\min_{x \in \mathcal{K}} \ v_t^\top x + \frac{H}{2}\|x\|_2^2$
  Sample $\tau_{t+1} \sim U[0,1]$
  Output $S_{t+1} = \{i : x_{t+1}(i) > \tau_t\}$ and observe f_{t+1}
  Compute $\nabla \hat{f}_t(x_{t+1})$ and pass $\nabla \hat{f}_{t+1}(x_{t+1})$ to TBAP($\{\nabla \hat{f}_i(x_i)\}, L, \epsilon$), and receive current partial sum v_{t+1}
end for

Algorithm 7 is differentially private (Theorem 17) and achieves $\tilde{O}(\sqrt{T})$ regret (Theorem 18).

Theorem 17 (Privacy guarantee)**.** SUBMODPFTAL($\{f_i\}_{i=1}^T, M, H, L, [n], \epsilon$) *is ϵ-differentially private for any sequence of functions $f_1, \ldots, f_T$ with bounded range $[-M, M]$ and for any $M, H, L, n, T > 0$.*

Proof. By Theorem 30 we know that the output of TBAP, $\{v_t\}_{t=1}^T$, is ϵ-differentially private. By Theorem 14 we get that the sequence $\{x_t\}_{t=1}^T$ is ϵ-differentially private since the

procedure $x_{t+1} \leftarrow \arg\min_{x \in K} v_t^\top x + \frac{H}{2}||x||_2^2$ is simply post-processing of the v_t's. Computing the output $\{S_t\}_{t=1}^T$ is further post-processing of the sequence $\{x_t\}_{t=1}^T$, and Theorem 14 again yields the result. □

Theorem 18 (Regret guarantee)**.** SUBMODPFTAL$(\{f_i\}_{i=1}^T, M, H, L, [n], \epsilon)$ *run with* $H = M\sqrt{T}$ *and* $||\nabla \hat{f}_t|| \leq L = 4M$ *for any sequence of submodular functions* $f_1, \ldots, f_T : 2^{[n]} \to [-M, M]$ *for any* $M, n, T > 0$ *guarantees,*

$$\mathbb{E}[\text{Regret}(T)] \leq O\left(\frac{nM^2 \ln^{1.5}(T)\sqrt{T}}{\epsilon}\right),$$

where the expectation is taken over the randomness of TBAP *and the sampling procedure used to choose* S_t*.*

Proof. Let $E_{TBAP}[\cdot]$ be the expectation taken with respect to the randomness of TBAP. Notice that

$$\begin{aligned}
\mathbb{E}[\sum_{t=1}^{T} f_t(S_t)] &= \mathbb{E}_{TBAP}[\mathbb{E}[\sum_{t=1}^{T} f_t(S_t)|TBAP]] \\
&= \mathbb{E}_{TBAP}[\mathbb{E}[\sum_{t=1}^{T-1} f_t(S_t)|TBAP]] \\
&\quad + \mathbb{E}_{TBAP,\tau_1,\ldots,\tau_{T-1}}[\mathbb{E}[f_T(S_T)|TBAP, \tau_1, ..., \tau_{T-1}]] \\
&= \mathbb{E}_{TBAP}[\mathbb{E}[\sum_{t=1}^{T-1} f_t(S_t)|TBAP]] + \mathbb{E}_{TBAP}[\hat{f}_T(x_T)]
\end{aligned}$$

by definition of $\hat{f}$. Repeating the argument $T-1$ more times we get $\mathbb{E}[\sum_{t=1}^T f_t(S_t)] = \mathbb{E}_{TBAP}[\sum_{t=1}^T \hat{f}_t(\hat{x}_t)]$. Now,

$$\begin{aligned}
&\mathbb{E}[\sum_{t=1}^{T} f_t(S_t) - \min_{S \subseteq [n]} \sum_{t=1}^{T} f_t(S)] \\
&\leq \mathbb{E}[\sum_{t=1}^{T} f_t(S_t) - \min_{x \in \mathcal{K}} \sum_{t=1}^{T} \hat{f}_t(x)]
\end{aligned}$$

$$
\begin{aligned}
&= \mathbb{E}_{TBAP}[\sum_{t=1}^{T} \hat{f}_t(x_t) - \min_{x \in \mathcal{K}} \sum_{t=1}^{T} \hat{f}_t(x)] \\
&\leq \mathbb{E}_{TBAP}[\sum_{t=1}^{T} \hat{f}_t(\hat{x}_t) - \min_{x \in \mathcal{K}} \sum_{t=1}^{T} \hat{f}_t(x)] + \frac{4nL^2T\ln^{1.5}(T)}{\epsilon H} \quad \text{(by Theorem 16)} \\
&\leq \frac{2}{H}\sum_{t=1}^{T} ||\nabla \hat{f}_t(\hat{x}_t)||^2 + \frac{H}{2}[||x||^2 - ||\hat{x}_1||^2] + \frac{4nL^2T\ln^{1.5}(T)}{\epsilon H} \quad \text{(by Theorem 15)} \\
&\leq \frac{2TL^2}{H} + \frac{Hn}{2} + \frac{4nL^2T\ln^{1.5}(T)}{\epsilon H}
\end{aligned}
$$

Plugging in the bound on L from Lemma 25 and choosing $H = M\sqrt{T}$ yields the result.

□

3.4 Bandit Setting

In this section we present Submodular Private Follow The Approximate Leader with Bandit Feedback (BANDITSUBMODPFTAL). This algorithm is differentially private and achieves a no regret guarantee for online submodular minimization with bandit feedback. The regret bound only loses a factor of $O(n^{1/2}\log^{1.5}(T))$ relative to the best known algorithm in the non-private setting.

The bandit setting makes the problem much more challenging because we do not have access to the whole function f_t nor to its subgradients. Instead we only observe the function evaluated at a single point, $f_t(S_t)$ for our chosen set S_t. This means that we can no longer compute subgradients of the Lovasz extension $\nabla \hat{f}_t$ and run RFTL on functions $\hat{f}_t$ as in the full information setting.

The key to obtaining sublinear regret is to balance exploration and exploitation. In this setting, exploitation is achieved by sampling S_t exactly from the distribution μ defined (through the Lovasz extension) by iterate x_t of BANDITSUBMODPFTAL.

However, if we sample according to the distribution over sets μ, we do not learn anything about the function's subgradients so, it is unclear what to do in future steps. To fix this, we should sample from some distribution that is close to μ, that allows us to explore

(i.e., obtain an unbiased estimate of the Lovasz extension at x_t). We use the sampling procedure from Hazan and Kale [68] to achieve this.

With these modifications, BANDITSUBMODPFTAL now works similarly to SUBMODPFTAL for the full information setting. The algorithm works by computing an unbiased estimator $\hat{g}_t$ of the gradient of the Lovasz extension $\nabla \hat{f}_t$, updating a private iterate $x_t \in \mathcal{K}$ using TBAP to obtain a private partial sum of $\sum_{j=1}^{t} \hat{g}_t$, and outputting a random set S_t that depends on x_t. We now present the full algorithm BANDITSUBMODPFTAL in Algorithm 8.

Algorithm 8 Submodular Private Regularized Follow The Leader with Bandit Feedback: BANDITSUBMODPFTAL($\{f_i\}_{i=1}^{T}, M, H, L, [n], \epsilon, \gamma$)

Input: Online sequence of submodular cost functions $\{f_1, ..., f_T\}$, lower and upper bounds function values $[-M, M]$, strong convexity parameter H, Lipschitz parameter L, ground set $[n]$, privacy parameter ϵ, parameter $\gamma \in (0, 1)$.
Output: Sequence of sets $S_1, \ldots, S_T \subseteq [n]$
Initialize $x_i \leftarrow \arg\min_{x \in \mathcal{K}} ||x||^2$
for t=1, . . . , T **do**
  Find maximal chain associated with x_t, $\emptyset = B_0 \subset B_1 \subset B_2 \subset \cdots B_n = [n]$, let π be the associated permutation
  Write x_t as $x_t = \sum_{i=0}^{n} \mu_i \mathcal{X}_{B_i}$, where $\mu_i = 0$ for the extra sets B_i that where added to complete the maximal chain for x_t.
  Sample S_t according to distribution: $S_t = B_i$ with probability $\rho_i = (1-\gamma)\mu_i + \frac{\gamma}{n+1}$
  Output S_t and observe $f_t(S_t)$
  if $S_t = B_0$ **then**
    Set $\hat{g}_t = -\frac{1}{\rho_0} f_t(B_0) e_{\pi^{-1}(1)}$
  else if $S_t = B_n$ **then**
    Set $\hat{g}_t = \frac{1}{\rho_n} f_t(B_n) e_{\pi^{-1}(n)}$
  else
    Choose $\xi \in \{+1, -1\}$ uniformly at random
    if $\xi = +1$ **then**
      Set $\hat{g}_t = \frac{2}{\rho_i} f_t(B_i) e_{\pi^{-1}(i)}$
    else
      Set $\hat{g}_t = -\frac{2}{\rho_i} f_t(B_i) e_{\pi^{-1}(i+1)}$
    end if
  end if
  Pass $\hat{g}_t$ to TBAP($\{\hat{g}_i\}, L, \epsilon$), and receive current partial sum $\hat{v}_t$
  Update $x_{t+1} = \arg\min_{x \in \mathcal{K}} \hat{v}_t^\top x + \frac{H}{2}||x||^2$
end for

In the algorithm e_i refers to the vector with all entries equal to 0 except for the i-th entry

which is equal to 1. The analysis of BANDITSUBMODPFTAL relies on the following key properties of the estimate $\hat{g}$.[1]

Lemma 26. *Let $\gamma \in (0,1)$. The random vector $\hat{g}_t$ computed in* BANDITSUBMODPFTAL *is an unbiased estimate of a subgradient of the Lovasz extension $\hat{f}_t$ of submodular f_t, evaluated at point x_t. That is,*

$$\mathbb{E}\left[\hat{g}_t \mid x_t\right] = \nabla \hat{f}_t(x_t).$$

Proof. Notice that conditioned on the randomness up to $t-1$

$$\hat{g}_t = \begin{cases} -\frac{1}{\rho_0} f_t(B_0) e(\pi^{-1}(1)) & \text{with probability } \rho_0 \\ \frac{2}{\rho_i} f_t(B_i) e(\pi^{-1}(i)) & \text{with probability } \frac{\rho_i}{2} \text{ for } 1 \leq i \leq n-1 \\ -\frac{2}{\rho_i} f_t(B_i) e(\pi^{-1}(i+1)) & \text{with probability } \frac{\rho_i}{2} \text{ for } 1 \leq i \leq n-1 \\ \frac{1}{\rho_n} f_t(B_n) e(\pi^{-1}(n)) & \text{with probability } \rho_n \end{cases}$$

Therefore

$$\begin{aligned} \mathbb{E}_t[\hat{g}_t] &= \rho_0[-\frac{1}{\rho_0} f_t(B_0) e(\pi^{-1}(1))] + \frac{\rho_1}{2}[\frac{2}{\rho_1} f_t(B_1) e(\pi^{-1}(1)) - \frac{2}{\rho_1} f_t(B_1) e(\pi^{-1}(2))] \\ &+ ... + \frac{\rho_{n-1}}{2}[\frac{2}{\rho_{n-1}} f_t(B_{n-1}) e(\pi^{-1}(n-1)) - \frac{2}{\rho_{n-1}} f_t(B_{n-1}) e(\pi^{-1}(n))] \\ &+ \rho_n[\frac{1}{\rho_n} f_t(B_n) e(\pi^{-1}(n))] \\ &= [f_t(B_1) - f_t(B_0)] e(\pi^{-1}(1)) + [f_t(B_2) - f_t(B_1)] e(\pi^{-1}(2)) \\ &+ ... + [f_t(B_n) - f_t(B_{n-1})] e(\pi^{-1}(n)) \end{aligned}$$

This means that $\mathbb{E}_t[\hat{g}_t](\pi^{-1}(i)) = f(B_i) - f_t(B_{i-1})$ for $i = 1, ..., n$. This concludes the proof since $\mathbb{E}_t[\hat{g}_t](i) = \mathbb{E}_t[\hat{g}_t](\pi^{-1}[\pi(i)]) = f_t(B_{\pi(i)}) - f_t(B_{\pi(i)-1}) = g_t(i)$ for $i = 1, ..., n$.

[1] Our Lemmas 26 and 27 were asserted without proof in [68]. Due to minor errors in the construction of $\hat{g}_t$ in [68], these claims are easily seen to be false under their construction. Here, we build the correct estimator and prove its correctness.

□

Lemma 27. *The random vector $\hat{g}_t$ computed in* BANDITSUBMODPFTAL *satisfies the following bound on its expected L_2-norm,*

$$\mathbb{E}\left[\|\hat{g}_t\|^2\right] \leq \frac{16M^2n^2}{\gamma},$$

where the expectation is taken over the algorithm's internal randomness up to time t.

Proof.

$$\begin{aligned}
\mathbb{E}_t[||\hat{g}_t||^2] &= \rho_0[-\frac{1}{\rho_0}f_t(B_0)]^2 + \sum_{i=1}^{n-1}\frac{\rho_i}{2}[(\frac{2}{\rho_i}f_t(B_i))^2 + (-\frac{2}{\rho_i})f_t(B_i)^2] + \rho_n[\frac{1}{\rho_n}f_t(B_n)^2] \\
&\leq 4M^2\sum_{i=0}^{n}\frac{1}{\rho_i} \\
&= 4M^2\sum_{i=0}^{n}\frac{1}{(1-\gamma)\mu_i + \gamma/(n+1)} \\
&= \sum_{i=0}^{n}\frac{n+1}{(1-\gamma)\mu_i(n+1)+\gamma} \\
&\leq \frac{4M^2(n+1)^2}{\gamma} \\
&\leq \frac{16M^2n^2}{\gamma}
\end{aligned}$$

The second to last inequality holds as long as $\gamma \leq 1$ which will be ensured when we choose the parameters of the algorithm. □

The exploration-exploitation dilemma can be better understood through the parameter γ. This parameter trades off between variance of the estimate $\hat{g}_t$ and the approximation of the Lovasz extension $\hat{f}_t$ to the true submodular function f_t. When γ is large, the variance of $\hat{g}_t$ is diminished, as can be seen in the statement of Lemma 27. When γ is small, the performance of $f_t(S_t)$ is close to that of $\hat{f}_t(x_t)$ (see Lemma 28 in Section 3.4.1). In the

statement of our main result (Theorem 20), we optimally tune γ to balance exploration and exploitation and minimize overall regret of BANDITSUBMODPFTAL.

Our two main results of this section show that BANDITSUBMODPFTAL is differentially private and achieves low regret.

Theorem 19 (Privacy guarantee)**.** BANDITSUBMODPFTAL$(\{f_i\}_{i=1}^T, M, H, L, [n], \epsilon, \gamma)$ *is ϵ-differentially private for any sequence of functions $f_1, \ldots, f_T$ with bounded range $[-M, M]$ and for any $M, H, L, n, T, \gamma > 0$.*

Proof. By Theorem 30 we know that the output of TBAP, $\{v_t\}_{t=1}^T$, is ϵ-differentially private. Notice that BANDITSUBMODPFTAL is running PFTAL on regularized functions $\hat{g}_t^\top x + \frac{H}{2}\|x\|^2$ thus by the same reasoning as in Theorem 17, the sequence $\{x_t\}_{t=1}^T$ is ϵ-differentially private since the procedure $x_{t+1} \leftarrow \arg\min_{x \in K} v_t^\top x + \frac{H}{2}\|x\|_2^2$ is simply post-processing of the v_t's. Since $\{S_t\}_{t=1}^T$ is post-processing on the sequence $\{x_t\}_{t=1}^T$, applying Theorem 14 again completes the proof. □

Theorem 20 (Regret guarantee)**.** BANDITSUBMODPFTAL$(\{f_i\}_{i=1}^T, M, H, L, [n], \epsilon, \gamma)$ *run with $H = MT^{2/3}$, $L = \frac{4Mn}{\sqrt{\gamma}}$, and $\gamma = \frac{n^{3/2}}{T^{1/3}}$ for any sequence of submodular functions $f_1, \ldots, f_T : 2^{[n]} \to [-M, M]$ for any $M, n, T > 0$ guarantees,*

$$\mathbb{E}[\text{Regret}(T)] \leq \tilde{O}\left(\frac{MnT^{2/3}}{\epsilon}\right),$$

where the expectation is taken with respect to all the internal randomness of the algorithm.

The proof of Theorem 20 relies on several key lemmas, presented in Section 3.4.1.

3.4.1 Regret Analysis of BANDITSUBMODPFTAL

There are several sources of potential sub-optimality in the output of BANDITSUBMODPFTAL that must be bounded. Firstly, the algorithm optimizes using continuous iterates x_t instead of discrete (Lemma 28). The algorithm incurs additional loss from the noise added

in TBAP to preserve privacy (Lemma 32). Lastly, due to the bandit feedback, we cannot compute an exact subgradient of the regularized Lovasz extension, and must instead use a (random) unbiased estimator (Lemmas 29 and 31).

The following lemmas bound the regret from these sources of error, and are used in the proof of Theorem 20.

We start with a lemma from Hazan and Kale [68], showing that the additional loss from choosing a subset of the ground set S_t instead of the point in $x_t \in \mathcal{K}$ is not too large.

Lemma 28 ([68])**.** *For any submodular function $f_t : [n] \to [-M, M]$, let x_t and S_t be the corresponding iterates and sets as defined in* BANDITSUBMODPFTAL*, then $\mathbb{E}[f_t(S_t)] \leq \mathbb{E}[\hat{f}_t(x_t)] + 2\gamma M$. Where the expectation is taken with respect to all the randomness of the algorithm.*

The proof is identical to that of [68]. We present it here for completeness. Let $\mathbb{E}_t$ be the expectation with respect to the randomness of the algorithm in round t conditioned on the history up to time $t-1$.

Proof. We know $\mathbb{E}_t[f_t(S_t)] = \sum_{i=0}^{n} \rho_i f_t(B_i)$ and $\hat{f}_t(x_t) = \sum_{i=0}^{n} \mu_i f(B_i)$. Therefore,

$$\begin{aligned}
\mathbb{E}_t[f_t(S_t)] - \hat{f}_t(x_t) &= \sum_{i=0}^{n} (\rho_i - \mu_i) f_t(B_i) \\
&\leq \gamma \sum_{i=0}^{n} \left[\frac{1}{n+1} + \mu_i \right] |f_t(B_i)| \\
&= \gamma \left(\frac{n}{n+1} + 1 \right) M \\
&\leq 2\gamma M.
\end{aligned}$$

Taking expectation with respect to the randomness up to time $t-1$ yields the result. □

The following lemma bounds the regret loss due to the fact that we only have bandit feedback. The main idea of the proof comes from [53], the first paper that provided an algorithm for online convex optimization with bandit feedback, however we must modify

it accordingly to account for the fact that our one-point gradient estimator is for the Lovasz extension of a submodular function and not just any convex function. This modification will exploit the bound on the variance of $\hat{g}_t$ from Lemma 27 and will allow us to prove a regret rate of $\tilde{O}(T^{2/3})$ instead of $\tilde{O}(T^{3/4})$ which is obtained for general convex functions while trying to preserve privacy (see [121]).

The next lemma bounds the loss our algorithm incurs due to bandit feedback against an adaptive adversary. The key to prove such a result is to bound with probability one the absolute difference between $\sum_{t=1}^{T} \nabla \hat{f}_t(x_t)$ and $\sum_{t=1}^{T} \nabla \hat{g}_t$, then use the fact that $\hat{g}_t$ is an unbiased estimator of $\nabla \hat{f}_t$.

Lemma 29. *Let $\{\hat{g}_t\}_{t=1}^{T}$ be the sequence of one point gradient estimates generated by* BANDITSUBMODPFTAL*$(\{f_i\}_{i=1}^{T}, M, H, L, [n], \epsilon, \gamma)$. Then,*

$$\mathbb{E}\left[\min_{x\in\mathcal{K}} \sum_{t=1}^{T} \hat{g}_t^\top x\right] \leq \mathbb{E}\left[\min_{x\in\mathcal{K}} \sum_{t=1}^{T} \nabla \hat{f}_t^\top x\right] + \frac{8Mn\sqrt{T}}{\sqrt{\gamma}},$$

where the expectation is taken with respect to all the randomness of the algorithm.

Proof. Define $\alpha_t = \nabla \hat{f}_t - \hat{g}_t$. Notice that with probability 1

$$\begin{aligned} &|\sum_{t=1}^{T} \hat{g}_t^\top x - \sum_{t=1}^{T} \nabla \hat{f}_t^\top x| \\ &\leq ||x||_2 || \sum_{t=1}^{T} \alpha_t||_2 \quad \text{(by Cauchy Schwartz)} \end{aligned}$$

Therefore, with probability 1

$$\min_{x\in\mathcal{K}} \sum_{t=1}^{T} \hat{g}_t^\top x \leq \min_{x\in\mathcal{K}} \sum_{t=1}^{T} \nabla \hat{f}_t(x_t)^\top x + || \sum_{t=1}^{T} \alpha_t||_2. \tag{3.1}$$

The previous ensures that our regret bound holds against adaptive adversaries.

We next proceed to bound $\mathbb{E}\left[\| \sum_{t=1}^{T} \alpha_t\|_2\right]^2$. By Lemma 30 stated below, $\mathbb{E}[\alpha_t^\top \alpha_{t'}] = 0$

for $t \neq t'$.

$$\begin{aligned}
\mathbb{E}\left[\left\|\sum_{t=1}^{T}\alpha_t\right\|_2\right]^2 &\leq \mathbb{E}\left[\left\|\sum_{t=1}^{T}\alpha_t\right\|_2^2\right] \qquad \text{(by Jensen's inequality)} \\
&= \sum_{t=1}^{T}\mathbb{E}\left[\|\alpha_t\|_2^2\right] + 2\sum_{t<t'}\mathbb{E}\left[\alpha_t^\top \alpha_{t'}\right] \\
&= \sum_{t=1}^{T}\mathbb{E}\left[\|\nabla \hat{f}_t(x_t) - \hat{g}_t\|_2^2\right] \\
&\leq \sum_{t=1}^{T}\mathbb{E}\left[2\|\nabla \hat{f}_t(x_t)\|_2^2 + 2\|\hat{g}_t\|_2^2\right] \\
&\leq 4T \cdot \frac{16M^2n^2}{\gamma}
\end{aligned}$$

where the last line follows from Lemma 27, and the fact that if $\|\hat{g}_t\|_2 \leq G$ then $\|\nabla \hat{f}_t(x_t)\|_2 \leq G$ by Jensen's inequality. Taking expectation on both sides of equation 3.1 yields the result □

The following lemma was asserted without proof in [121]. We prove it here for completeness.

Lemma 30. *Let $\alpha_t = \nabla \hat{f}_t(x_t) - \hat{g}_t$. Then, for $t < t'$ it holds that $\mathbb{E}[\alpha_t^\top \alpha_{t'}] = 0$, where the expectation is taken over the randomization of the algorithm used to build the estimates of the gradient $\{\hat{g}_t\}_{t=1}^T$.*

Proof.

$$\begin{aligned}
\mathbb{E}[\alpha_t^\top \alpha_{t'}] &= \mathbb{E}[(\nabla \hat{f}_t(x_t) - \hat{g}_t)^\top (\nabla \hat{f}_{t'}(x_{t'}) - \hat{g}_{t'})] \\
&= \mathbb{E}[\nabla \hat{f}_t(x_t)^\top \nabla \hat{f}_{t'}(x_{t'})] - \mathbb{E}[\nabla \hat{f}_t(x_t)^\top \hat{g}_{t'}] - \mathbb{E}[\nabla \hat{f}_{t'}(x_{t'})^\top \hat{g}_t] + \mathbb{E}[\hat{g}_t^\top \hat{g}_{t'}] \\
&= \nabla \hat{f}_t(x_t)^\top \nabla \hat{f}_{t'}(x_{t'}) - \nabla \hat{f}_t(x_t)^\top \nabla \hat{f}_{t'}(x_{t'}) - \nabla \hat{f}_{t'}(x_{t'})^\top \nabla \hat{f}_t(x_t) + \mathbb{E}[\hat{g}_t^\top \hat{g}_{t'}]
\end{aligned}$$

We now show that $\mathbb{E}[\hat{g}_t^\top \hat{g}_{t'}] = \nabla \hat{f}_{t'}(x_{t'})^\top \nabla \hat{f}_t(x_t)$.

$$\begin{aligned}
\mathbb{E}[\hat{g}_t^\top \hat{g}_{t'}] &= \mathbb{E}_{1,\dots t'-1}[\mathbb{E}_{t'}[\hat{g}_t^\top \hat{g}_{t'} | t = 1, \dots t' - 1]] \\
&= \mathbb{E}_{1,\dots t'-1}[\hat{g}_t^\top \mathbb{E}_{t'}[\hat{g}_{t'} | t = 1, \dots t' - 1]] \\
&= \mathbb{E}_{1,\dots t'-1}[\hat{g}_t^\top \nabla \hat{f}_{t'}(x_{t'})] \\
&= \nabla \hat{f}_t^\top(x_t) \nabla \hat{f}_{t'}(x_{t'})
\end{aligned}$$

□

Lemma 31. *Let $\{\hat{g}_t\}_{t=1}^T$ and $\{x_t\}_{t=1}^T$ be the sequences generated by* BANDITSUBMODPFTAL*$(\{f_i\}_{i=1}^T, M, H, L, [n], \epsilon, \gamma)$. Then,*

$$\mathbb{E}[\sum_{t=1}^T \hat{g}_t^\top x_t] = \mathbb{E}[\sum_{t=1}^T \nabla \hat{f}_t^\top x_t],$$

where the expectation is taken with respect to all the randomness of the algorithm.

Proof.

$$\begin{aligned}
&\mathbb{E}[\sum_{t=1}^T \hat{g}_t^\top x_t] \\
&= \mathbb{E}[\sum_{t=1}^{T-1} \hat{g}_t^\top x_t] + \mathbb{E}[\hat{g}_T^\top x_T] \\
&= \mathbb{E}[\sum_{t=1}^{T-1} \hat{g}_t^\top x_t] + \mathbb{E}[\mathbb{E}[\hat{g}_T^\top x_T | \tau = 1, \dots, T-1]] \\
&= \mathbb{E}[\sum_{t=1}^{T-1} \hat{g}_t^\top x_t] + \mathbb{E}[x_T^\top \mathbb{E}[\hat{g}_T | \tau = 1, \dots, T-1]] \\
&= \mathbb{E}[\sum_{t=1}^{T-1} \hat{g}_t^\top x_t] + \mathbb{E}[x_T^\top \mathbb{E}[\hat{g}_T | \tau = 1, \dots, T-1]] \\
&= \mathbb{E}[\sum_{t=1}^{T-1} \hat{g}_t^\top x_t] + \mathbb{E}[x_T^\top \nabla \hat{f}_T] \quad \text{by Lemma 26.}
\end{aligned}$$

Repeating the argument $T - 1$ more times yields the result. □

The following lemma quantifies the loss in the regret due to privacy.

Lemma 32. *Let $\{x_t\}_{t=1}^T$ be the sequence generated by* BANDITSUBMODPFTAL*$(\{f_i\}_{i=1}^T, M, H, L, [n], \epsilon, \gamma)$. Let $\hat{x}_t$ be the non private iterate of the algorithm, that is $\hat{x}_{t+1} = \sum_{\tau=1}^t \hat{g}_\tau^\top x + \frac{H}{2}||x||^2$. Then,*

$$\mathbb{E}[\sum_{t=1}^T \hat{g}_t^\top x_t] \leq \mathbb{E}[\sum_{t=1}^T \hat{g}_t^\top \hat{x}_t] + \frac{64n^3M^2T\ln^{1.5}(T)}{\epsilon\gamma H},$$

where the expectation is taken with respect to the randomness of the algorithm.

Proof. We follow the proof of Lemma 8 in [121].

Let $J_t = v_t^\top x + \frac{H}{2}||x||^2$. Let $\xi_t = v_t - \sum_{\tau=1}^t \hat{g}_t$ be the noise added by TBAP to $\sum_{\tau=1}^t \hat{g}_t$. Notice that $x_{t+1} = \arg\min_{x\in\mathcal{K}} J_t(x) + \xi_t^\top x$ and $\hat{x}_t = \arg\min_{x\in\mathcal{K}} J_t(x)$. Since J_t is H-strongly convex we have that

$$||\hat{x}_{t+1} - x_{t+1}|| \leq \frac{2||\xi_t||}{H}.$$

Since each ξ_t is formed in TBAP by adding at most $\lceil \ln(T)+1 \rceil$ vectors with norms drawn from a Gamma distribution with scale n and shape $\frac{(\lceil \ln(T)+1 \rceil)4Mn}{\sqrt{\gamma}\epsilon}$ we can upper bound $\mathbb{E}[||\xi_t||]$ by $\frac{16n\ln^{1.5}(T)Mn}{\epsilon\sqrt{\gamma}}$.

Since $\hat{g}_t^\top$ is $\frac{4Mn}{\sqrt{\gamma}}$-Lipschitz continuous (by Lemma 27, concavity of $\sqrt{\cdot}$, and Jensen's inequality) we have that,

$$\begin{aligned}\mathbb{E}[\sum_{t=1}^T \hat{g}_t^\top x_t] &\leq \mathbb{E}[\sum_{t=1}^T \hat{g}_t^\top \hat{x}_t] + \mathbb{E}[\frac{4Mn}{\sqrt{\gamma}}\sum_{t=1}^T \frac{2||\xi_t||}{H}] \\ &\leq \mathbb{E}[\sum_{t=1}^T \hat{g}_t^\top \hat{x}_t] + \frac{64n^3M^2T\ln^{1.5}(T)}{\epsilon\gamma H}.\end{aligned}$$

□

We are now ready to prove the regret guarantee of BANDITSUBMODPFTAL.

Proof of Theorem 20.

$$
\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T} f_t(S_t) - \min_{S\subseteq[n]} \sum_{t=1}^{T} f_t(S)\right] \\
&\leq \mathbb{E}\left[\sum_{t=1}^{T} f_t(S_t) - \min_{x\in\mathcal{K}} \sum_{t=1}^{T} \hat{f}_t(x)\right] \\
&\leq \mathbb{E}\left[\sum_{t=1}^{T} \hat{f}_t(x_t) - \sum_{t=1}^{T} \hat{f}_t(x)\right] + 2\gamma MT \quad \text{for any } x \in \mathcal{K} \text{ by Lemma 28} \\
&\leq \mathbb{E}\left[\sum_{t=1}^{T} \nabla \hat{f}_t^\top (x_t - x)\right] + 2\gamma MT \quad \text{since } \hat{f}_t \text{ is convex} \\
&\leq \mathbb{E}\left[\sum_{t=1}^{T} \nabla \hat{f}_t^\top x_t\right] - \mathbb{E}\left[\min_{x\in\mathcal{K}} \sum_{t=1}^{T} \hat{g}_t^\top x\right] + 2\gamma MT + \frac{8Mn\sqrt{T}}{\sqrt{\gamma}} \quad \text{by Lemma 29} \\
&= \mathbb{E}\left[\sum_{t=1}^{T} \hat{g}_t^\top x_t\right] - \mathbb{E}\left[\min_{x\in\mathcal{K}} \sum_{t=1}^{T} \hat{g}_t^\top x\right] + 2\gamma MT + \frac{8Mn\sqrt{T}}{\sqrt{\gamma}} \quad \text{by Lemma 31} \\
&= \mathbb{E}\left[\sum_{t=1}^{T} \hat{g}_t^\top \hat{x}_t\right] - \mathbb{E}\left[\min_{x\in\mathcal{K}} \sum_{t=1}^{T} \hat{g}_t^\top x\right] + 2\gamma MT + \frac{8Mn\sqrt{T}}{\sqrt{\gamma}} + \frac{64n^3M^2T\ln^{1.5}(T)}{\epsilon\gamma H} \\
&\quad \text{by Lemma 32} \\
&\leq \mathbb{E}\left[\frac{2}{H}\sum_{t=1}^{T} ||\hat{g}_t||_2^2 + \frac{H}{2}[||x||_2^2 - ||x_1||_2^2]\right] + 2\gamma MT + \frac{8Mn\sqrt{T}}{\sqrt{\gamma}} + \frac{64n^3M^2T\ln^{1.5}(T)}{\epsilon\gamma H} \\
&\quad \text{for any } x \in \mathcal{K} \text{ by Theorem 15} \\
&\leq \frac{32M^2n^2T}{H\gamma} + nH + 2\gamma MT + \frac{8Mn\sqrt{T}}{\sqrt{\gamma}} + \frac{64n^3M^2T\ln^{1.5}(T)}{\epsilon\gamma H} \quad \text{by Lemma 27}
\end{aligned}
$$

Choosing $\gamma = \frac{n^{3/2}}{T^{1/3}}$, $H = MT^{2/3}$ yields the result. □

CHAPTER 4
COMPETING AGAINST EQUILIBRIA IN ZERO-SUM GAMES WITH EVOLVING PAYOFFS

4.1 Introduction

We consider a problem in which two players interact in a zero-sum game repeatedly. The payoff matrix of the game is unknown to the players *a priori*, and may change arbitrarily on each round. Our objective is to find competitive strategies that can achieve the Nash equilibrium of the game with the average payoffs in the long term. This problem is a significant extension of the classical learning setting in zero-sum games, where the underlying payoff matrix is often assumed to be fixed or i.i.d. In contrast, we allow the payoff matrix to evolve arbitrarily in each round, and can even be selected in a possibly adversarial fashion.

Zero-sum games [101, 128] are ubiquitous in economics and central to understanding Linear Programming duality [8, 72], convex optimization [4, 5], robust optimization [23], and Differential Privacy [47]. The task of finding the Nash equilibrium of a zero-sum game is also connected to several machine learning problems such as: Markov Games [94], Boosting [56], Multiarmed Bandits with Knapsacks [18, 76] and dynamic pricing problems [51].

We formally define the problem setting in Section 4.1.1. We then highlight the main contributions of this work in Section 4.1.2 and discuss related works in Section 4.1.3.

4.1.1 Problem Formulation: Online Matrix Games

We start by reviewing the definition of classical two-player zero-sum games. Suppose player 1 has d_1 possible actions and player 2 has d_2 possible actions. The payoffs for both players are determined by a matrix $A \in \mathbb{R}^{d_1 \times d_2}$, with $A_{i,j}$ corresponding to the loss of

player 1 and the reward of player 2 when they choose to play actions $(i, j) \in [d_1] \times [d_2]$.[1] We allow the players to use *mixed strategies* – each mixed strategy is represented by a probability distribution over their actions. More specifically, when Player 1 uses a mixed strategy $x \in \Delta_{d_1}$ and Player 2 uses a mixed strategy $y \in \Delta_{d_2}$, the expected payoff is $x^\top A y$.[2] Throughout the chapter, we refer to the static zero-sum game as a *matrix game* (MG), because the players' payoffs are a bilinear function encoded by the matrix A. A Nash equilibrium of this game is defined as any pair of (possibly) mixed strategies (x^*, y^*) such that

$$(x^*)^\top A y \leq (x^*)^\top A y^* \leq x^\top A y^*$$

for any $x \in \Delta_{d_1}, y \in \Delta_{d_2}$. It is well known that every MG has at least one Nash equilibrium [101]. The problem of finding an equilibrium for a MG can be reduced to solving linear programming problems. In fact, [8] showed that the opposite is also true, every linear programming problem can be solved by finding an equilibrium to a corresponding MG.

Now, we define a problem that generalizes the matrix games into an online setting, which we call the Online Matrix Games (OMG) problem. Suppose two players interact in a repeated zero-sum matrix game through T rounds. In every round $t \in [T]$, they must each choose a (possibly) mixed strategy from the given action sets $x_t \in \Delta_{d_1}, y_t \in \Delta_{d_2}$. However, we assume that the payoff matrix in OMG can evolve in each round, and the players have no knowledge of the payoff quantities in that round before they commit to an action. Let $\{A_t\}_{t=1}^T$ be an arbitrary sequence of matrices, where each $A_t \in [-1, 1]^{d_1 \times d_2}$ for all $t = 1, .., T$. For each round t, the players choose their mixed strategies $x_t \in \Delta_{d_1}, y_t \in \Delta_{d_2}$ before the matrix A_t is revealed. Then, player 1 (resp. player 2) receives a loss (resp. gain) given by the payoff quantity $x_t^\top A_t y_t$. Note that the payoff matrix A_t is allowed to change arbitrarily from round to round and may even depend on the past actions of both players. The *joint goal* for both players is to find strategies that ensure their average payoffs

[1] Throughout, $[n] \triangleq \{1, ..., n\}$ for any positive integer n.

[2] Here, Δ_d represents the unit simplex in dimension d: $\Delta_d \triangleq \{v \in \mathbb{R}^d : \|v\|_1 = 1, v \geq 0\}$.

in T rounds is close to the Nash Equilibrium under the average payoff matrix $\frac{1}{T}\sum_{t=1}^{T} A_t$ in hindsight.

More precisely, let us call the quantity

$$\left| \sum_{t=1}^{T} x_t^\top A_t y_t - \min_{x \in X} \max_{y \in Y} \sum_{t=1}^{T} x^\top A_t y \right| \tag{4.1}$$

the *Nash Equilibrium (NE) regret*. This is a natural extension of the regret concept in typical online learning or multi-armed bandit problems, which involve only a single decision maker. The primary objective of the OMG problem is to find online strategies for both players so that, as $T \to \infty$, the average NE regret (4.1) per round tends to 0 (i.e., the NE regret is $o(T)$).

We make some remarks about the choice of benchmark and the fact that the players must update jointly despite the fact that they are playing a zero-sum game. In the following examples, the comparator term $\min_{x \in X} \max_{y \in Y} \sum_{t=1}^{T} x^\top A_t y$ arises naturally and there is one decision maker which chooses the actions of both players.

1. Online Linear Programming [11]: the decision maker solves an LP where data arrives sequentially. This problem has real-world applications in ad-auctions. Using Lagrangian duality, we can reduce this problem to an online zero-sum game (our setting), where player 1 chooses primal variables and player 2 chooses dual variables. Our benchmark corresponds to the optimal solution of the offline LP.

2. Adversarial Bandits with Knapsacks [76]: this problem extends the classical Multi Armed Bandit by adding a 'knapsack' constraint. Again, using a Lagrangian relaxation on the knapsack constraint, this problem can be linked to the online min-max games that we study (see Sec. 3.2 of [76]).

3. Generative Adversarial Networks [63]: GANs can also be viewed as a zero-sum game, where the decision maker trains the generator and discriminator to find a Nash equilibrium. Although our model cannot directly be used for GANs because they are

nonconvex, it is another example where both players may desire to update jointly. In Section 4.7 we explore this further.

In the chapter, we consider the OMG problem in two distinct information feedback settings. In the *full information* setting (Section 4.4), both players are able to observe the full matrix A_t at the end of round t. In the *bandit setting* (Section 4.5), players can only observe the entry of A_t indexed by (i_t, j_t) at the end of round t, where i_t and j_t are the actions sampled from the probability distributions associated with their mixed strategies (x_t, y_t).

4.1.2 Main Results

In addition to introducing a novel problem setting, the main contributions of the present work are as follows.

- First, we show that a natural "naïve" approach, where each player simply aims to minimize their individual regret, will fail to produce a sublinear NE regret algorithm, in the sense of (4.1), regardless of the players' no-regret strategies (Theorem 21).
- Second, in the full information setting, we provide an algorithm for the OMG problem that achieves a NE regret of $O(\max\{\ln(d_1), \ln(d_2)\}\ln(T)\sqrt{T})$ (Theorem 23). Note that the regret depends logarithmically on the number of actions, allowing us to handle scenarios where the players have exponentially many actions available. We also show a NE regret bound for general convex-concave games of $O(\max\{d_1, d_2\}\ln(T)\sqrt{T})$ (Theorem 22).
- Third, we propose an algorithm for the bandit setting that achieves a NE regret of order $O((\max\{d_1, d_2\})^{5/3}T^{5/6})$ (Theorem 25).
- Fourth, we extend our results to the case where the payoff function is strongly convex-concave. In this regime we are able to show a faster NE regret rate that scales logarithmically with respect to the number of rounds T (Theorem 26).

- Fifth, we show empirically how our algorithm can be used to prevent mode collapse when training GANs in a basic setup (Section 4.7).

4.1.3 Related Work

The reader familiar with Online Convex Optimization (OCO) may find it closely related to the OMG problem. In the OCO setting, a player is given a convex, closed, and bounded action set X, and must repeatedly choose an action $x_t \in X$ before the convex function $f_t(x) : X \to \mathbb{R}$ is revealed. The player's goal is to obtain sublinear *individual regret* defined as $\sum_{t=1}^{T} f_t(x_t) - \min_{x \in X} \sum_{t=1}^{T} f_t(x)$. This problem is well studied and several algorithms such as Online Gradient Descent [139], Regularized Follow the Leader [6, 116] and Perturbed Follow the Leader [82] achieve optimal individual regret bounds that scale as $O(\sqrt{T})$. The most natural (although incorrect) approach to attack the OMG problem is to equip each of the players with a sublinear individual regret algorithm. However, we will show in Section 4.3 that if both players use an algorithm that guarantees sublinear individual regret, then it is impossible to achieve sublinear NE regret when the payoff matrices are chosen adversarially. In other words, the algorithms for the OCO setting cannot be directly applied to the OMG problem considered in this work.

We now discuss some related works that focus on learning in games. [120] study a two player, two-action general sum static game. They show that if both players use Infinitesimal Gradient Ascent, either the strategy pair will converge to a Nash Equilibrium (NE), or even if they do not, then the average payoffs are close to that of the NE. A result of similar flavor was derived in [37] for any zero-sum convex-concave game. Given a payoff function $\mathcal{L}(x, y)$, they show that if both players minimize their individual-regrets, then the average of actions $(\bar{x}, \bar{y})$ will satisfy $|\mathcal{L}(\bar{x}, \bar{y}) - \mathcal{L}(x^*, y^*)| \to 0$ as $T \to \infty$, where (x^*, y^*) is a NE. [29] improve upon the result of [120] by proposing an algorithm called WoLF (Win or Learn Fast), which is a modification of gradient ascent; they show that the iterates of their algorithm indeed converge to a NE. [41] further improve the results in [120] and

[28] by developing an algorithm called GIGA-WoLF for multi-player nonzero sum static games. Their algorithm learns to play optimally against stationary opponents; when used in self-play, the actions chosen by the algorithm converge to a NE. More recently, [20] studied general multi-player static games and show that by decomposing and classifying the second order dynamics of these games, one can prevent cycling behavior to find NE. We note that unlike our work, all of the papers above consider repeated games with a static payoff matrix, whereas we allow the payoff matrix to change arbitrarily. An exception is the work by [74], who consider the same setting as our OMG problem; however their paper only shows that the sum of the individual regrets of both players is sublinear and does not study convergence to NE.

Related to the OMG problem with bandit feedback is the seminal work of [53]. They provide the first sublinear regret bound for Online Convex Optimization with bandit feedback, using a one-point estimate of the gradient. The one-point gradient estimate used in [53] is similar to those independently proposed in [64] and in [122]. The regret bound provided in [53] is $O(T^{3/4})$, which is suboptimal. In [6], the authors give the first $O(\sqrt{T})$ bound for the special case when the functions are linear. More recently, [70] and [33] designed the first efficient algorithms with $\tilde{O}(poly(d)\sqrt{T})$ regret for the general online convex optimization case; unfortunately, the dependence on the dimension d in the regret rate is a very large polynomial. Our one-point matrix estimate is most closely related to the random estimator in [14] for linear functions. It is possible to use the more sophisticated techniques from [6, 33, 70] to improve our NE regret bound in section 4.5; however, the result does not seem to be immediate and we leave this as future work.

4.2 Preliminaries

In this section we introduce notation and definitions that will be used throughout the chapter.

4.2.1 Notation

By default, all vectors are column vectors. A vector with entries $x_1, ..., x_d$ is written as $x = [x_1; ...; x_d] = [x_1, ..., x_d]^\top$, where $\top$ denotes the transpose. For a matrix A, let A_{ij} be the entry in the i-th row and j-th column.

4.2.2 Convex Functions

For any $H > 0$ we say that a function $f : X \to \mathbb{R}$ is H-strongly convex with respect to a norm $\|\cdot\|$, if for any $x_1, x_2 \in X$, it holds that

$$f(x_1) \geq f(x_2) + \nabla f(x_2)^\top (x_1 - x_2) + \frac{H}{2}\|x_1 - x_2\|^2.$$

Here, $\nabla f(x)$ denotes any subgradient of f at x. Strong convexity implies that the optimization problem $\min_{x \in X} f(x)$ has a unique solution. If $H = 0$ we simply say that the function is convex. We say a function g is H-strongly concave if $-g$ is H-strongly convex. Furthermore, we say a function $\mathcal{L}(x, y)$ is H-strongly convex-concave if for any fixed $y_0 \in Y$, the function $\mathcal{L}(x, y_0)$ is H-strongly convex in x, and for any fixed $x_0 \in X$, the function $\mathcal{L}(x_0, y)$ is H-strongly concave in y.

4.2.3 Saddle Points and Nash Equilibra

A pair (x^*, y^*) is called a saddle point for $\mathcal{L} : X \times Y \to \mathbb{R}$ if for any $x \in X$ and $y \in Y$, we have

$$\mathcal{L}(x^*, y) \leq \mathcal{L}(x^*, y^*) \leq \mathcal{L}(x, y^*). \tag{4.2}$$

It is well known that if $\mathcal{L}$ is convex-concave, and X and Y are convex and compact sets, there always exists at least one saddle point [see e.g. 30]. Moreover, if $\mathcal{L}$ is strongly convex-concave, the saddle point is unique.

A saddle point is also known as a Nash equilibrium for two-player zero-sum games [102]. In a matrix game, the payoff function $\mathcal{L}(x, y) = x^\top A y$ is bilinear, and therefore is

convex-concave. The action spaces of the two players are $X = \Delta_{d_1}$ and $Y = \Delta_{d_2}$, which are convex and compact. As a result, there always exists a Nash equilibrium for any matrix game. The famous von Neumann minimax theorem states that $\min_{x \in \Delta_{d_1}} \max_{y \in \Delta_{d_2}} x^\top A y = \max_{y \in \Delta_{d_2}} \min_{x \in \Delta_{d_1}} x^\top A y$. If Player 1 chooses $x^* \in \arg\min_{x \in \Delta_{d_1}} \max_{y \in \Delta_{d_2}} x^\top A y$ and Player 2 chooses $y^* \in \arg\max_{y \in \Delta_{d_2}} \min_{x \in \Delta_{d_1}} x^\top A y$, the pair (x^*, y^*) is an equilibrium of the game [101].

4.2.4 Lipschitz Continuity

We say a function $f : X \to \mathbb{R}$ is G-Lipschitz continuous with respect to a norm $\|\cdot\|$ if for all $x, y \in X$ it holds that

$$|f(x) - f(y)| \leq G\|x - y\|$$

It is well known that the previous inequality holds if and only if

$$\|\nabla f(x)\|_* \leq G$$

for all $x \in X$, where $\|\cdot\|_*$ denotes the dual norm of $\|\cdot\|$ [30, 117]. Similarly, we say a function $\mathcal{L}(x, y)$ is G-Lipschitz continuous with respect to a norm $\|\cdot\|$ if

$$|\mathcal{L}(x_1, y_1) - \mathcal{L}(x_2, y_2)| \leq G\|[x_1; y_1] - [x_2; y_2]\|.$$

for any $x_1, x_2 \in X$ and any $y_1, y_2 \in Y$. Again, the previous inequality holds if and only if

$$\|[\nabla_x \mathcal{L}(x, y); \nabla_y \mathcal{L}(x, y)]\|_* \leq G$$

for all $x \in X$, $y \in Y$.

Lemma 33. *Consider a matrix A. If the absolute value of each entry of A is bounded by*

$c > 0$, *then the function* $\mathcal{L}(x, y) = x^\top A y$ *is* $G_{\mathcal{L}}^{\|\cdot\|_2}$*-Lipschitz continuous with respect to* $\|\cdot\|_2$*, where* $G_{\mathcal{L}}^{\|\cdot\|_2} = \sqrt{c}\left(\sqrt{d_1} + \sqrt{d_2}\right)$*. The function* $\mathcal{L}$ *is also* $G_{\mathcal{L}}^{\|\cdot\|_1}$*-Lipschitz continuous with respect to norm* $\|\cdot\|_1$*, where* $G_{\mathcal{L}}^{\|\cdot\|_1} = c$*.*

Proof of Lemma 33. We omit the subscript t.

$$
\begin{aligned}
\|\nabla x^\top A y\|_2 &= \left\| \begin{bmatrix} \nabla_x x^\top A y \\ \nabla_y x^\top A y \end{bmatrix} \right\|_2 \\
&= \left\| \begin{bmatrix} A_{[1,:]}^\top y \\ \dots \\ A_{[d_1,:]}^\top y \\ A_{[:,1]}^\top x \\ \dots \\ A_{[:,d_2]}^\top x \end{bmatrix} \right\|_2 \\
&\leq \left\| \begin{bmatrix} A_{[1,:]}^\top y \\ \dots \\ A_{[d_1,:]}^\top y \end{bmatrix} \right\|_2 + \left\| \begin{bmatrix} A_{[:,1]}^\top x \\ \dots \\ A_{[:,d_2]}^\top x \end{bmatrix} \right\|_2 \\
&\leq \sqrt{\sum_{i=1}^{d_1} (A_{[i,:]}^\top y)^2} + \left\| \begin{bmatrix} A_{[:,1]}^\top x \\ \dots \\ A_{[:,d_2]}^\top x \end{bmatrix} \right\|_2 \\
&\leq \sqrt{d_1 (\|A_{[i,:]}\|_\infty \|y\|_1)^2} + \left\| \begin{bmatrix} A_{[:,1]}^\top x \\ \dots \\ A_{[:,d_2]}^\top x \end{bmatrix} \right\|_2 \qquad \text{by Generalized Cauchy Schwartz} \\
&\leq \sqrt{c d_1} + \left\| \begin{bmatrix} A_{[:,1]}^\top x \\ \dots \\ A_{[:,d_2]}^\top x \end{bmatrix} \right\|_2 \\
&\leq \sqrt{c d_1} + \sqrt{c d_2}. \quad \text{(using the same reasoning)}
\end{aligned}
$$

The second part of the claim follows by bounding $\|\nabla x^\top Ay\|_\infty$ using the same argument.

□

4.3 Challenges of the OMG Problem: An Impossibility Result

Recall that we defined the Online Matrix Games (OMG) problem in Section 4.1.1, where two players play a zero-sum games for T rounds. The sequence of payoff matrices $\{A_t\}_{t=1}^T$ is selected arbitrarily. In each round $t \in [T]$, both players choose their strategies before the payoff matrix A_t is revealed. The goal is to find strategies under which the players' average payoffs are close to the Nash Equilibrium of the game with payoff matrix $\sum_{t=1}^T A_t$.

Perhaps the most natural (albeit futile) approach to attack the OMG problem is to equip each of the players with a sublinear individual regret algorithm to generate a sequence of iterates $\{x_t, y_t\}_{t=1}^T$. We gave a few examples of Online Convex Optimization (OCO) algorithms that guarantee $O(\sqrt{T})$ regret in Section 4.1.3. However, if each player minimizes its individual regret greedily using OCO, this approach only implies that $\sum_{t=1}^T x_t^\top A_t y_t - \min_{x\in\Delta_X} \sum_{t=1}^T xA_t y_t = O(\sqrt{T})$, and $\max_{y\in\Delta_Y} \sum_{t=1}^T x_t^\top A_t y - \sum_{t=1}^T x_t^\top A_t y_t = O(\sqrt{T})$. Notice that the quantity $\min_{x\in X} \max_{y\in Y} \sum_{t=1}^T x^\top A_t y$ associated with the Nash Equilibrium in equation (4.1) does not even appear in these bounds. The reader familiar with saddle point computation may wonder how the so-called 'duality gap' [34]:
$\max_{y\in\Delta_Y} \sum_{t=1}^T x_t^\top A_t y - \min_{x\in\Delta_X} \sum_{t=1}^T xA_t y_t = O(\sqrt{T})$ relates to achieving sublinear NE regret. It is easy to see that the duality gap is the sum of individual regret of both players. In view of Theorem 21 we will see that NE regret and the duality gap are in some sense incompatible.

In this section we present a result that shows that there is no algorithm that *simultaneously* achieves sublinear NE regret and individual regret for both players. This implies that if both players individually use any existing algorithm from OCO they would inevitably fail to solve the OMG problem.

Theorem 21. *Consider any algorithm that selects a sequence of x_t, y_t pairs given the past*

payoff matrices $A_1, \ldots, A_{t-1}$. Consider the following three objectives:

$$\left| \sum_{t=1}^{T} x_t^\top A_t y_t - \min_{x \in \Delta} \max_{y \in \Delta} \sum_{t=1}^{T} x^\top A_t y \right| = o(T), \tag{4.3}$$

$$\sum_{t=1}^{T} x_t^\top A_t y_t - \min_{x \in \Delta_X} \sum_{t=1}^{T} x^\top A_t y_t = o(T), \tag{4.4}$$

$$\max_{y \in \Delta_Y} \sum_{t=1}^{T} x_t^\top A_t y - \sum_{t=1}^{T} x_t^\top A_t y_t = o(T). \tag{4.5}$$

Then there exists an (adversarially-chosen) sequence $A_1, A_2, \ldots$ such that not all of (4.3)*,* (4.4)*, and* (4.5)*, are true.*

A full proof of the result is shown in the next subsection, but here we give a sketch. The main idea is to construct two parallel scenarios, each with their own sequences of payoff matrices. The two scenarios will be identical for the first $T/2$ periods but are different for the rest of the horizon. In our particular construction, in both scenarios the players play the well known "matching-pennies" game for the first $T/2$ periods, then in first scenario they play a game with equal payoffs for all of their actions and in the second scenario they play a game where Player 1 is indifferent between its actions. One can show that if all three quantities in the statement of the theorem are $o(T)$ in the first scenario, then we prove that at least one of them is $\Omega(T)$ in the second one which yields the result. This suggests that the machinery for OCO, which minimizes individual regret, cannot be directly applied to the OMG problem.

4.3.1 Proof of Theorem 21

Next we give a formal proof of Theorem 21.

Proof of Theorem 21. We assume there exists an algorithm such that

$|\sum_{t=1}^{T} x_t^\top A_t y_t - \min_{x \in X} \max_{y \in Y} \sum_{t=1}^{T} x^\top A_t y| \leq o(T)$,

$\sum_{t=1}^{T} x_t^\top A_t y_t - \min_{x \in \Delta_X} \sum_{t=1}^{T} x^\top A_t y_t \leq o(T)$,

and $\max_{y\in\Delta_Y}\sum_{t=1}^{T}x_t^\top A_T y-\sum_{t=1}^{T}x_t^\top A_t y_t\leq o(T)$ for all possible sequences of matrices $\{A_t\}_{t=1}^T$ with bounded entries between $[-1,1]$. We now construct two sequences of functions for which all the three guarantees hold and lead that to a contradiction. Let T be divisible by 2. In scenario 1: $A_t=\begin{bmatrix}1 & -1\\ -1 & 1\end{bmatrix}$ for $1\leq t\leq\frac{T}{2}$ and $A_t=\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}$ for $\frac{T}{2}<t\leq T$. In scenario 2: $A_t=\begin{bmatrix}1 & -1\\ -1 & 1\end{bmatrix}$ for $1\leq t\leq\frac{T}{2}$ and $A_t=\begin{bmatrix}1 & -1\\ 1 & -1\end{bmatrix}$ for $\frac{T}{2}<t\leq T$. It is easy to see that for both scenarios it holds that $\min_{x\in X}\max_{y\in Y}\sum_{t=1}^{T}x^\top A_t y=0$. Since $d_1=d_2=2$ and we can parametrize any $x\in\Delta_X$ as $x=[\alpha;1-\alpha]$ and any $y\in\Delta_Y$ as $y=[\beta;1-\beta]$ for some $0\leq\alpha,\beta\leq 1$. By assumption we have that $\max_{y\in\Delta_Y}\sum_{t=1}^{T}x_t^\top A_t y-\min_{x\in X}\max_{y\in Y}\sum_{t=1}^{T}x^\top A_t y\leq o(T)$ for all sequences of matrices $\{A_t\}_{t=1}^T$. This implies for scenario 1 that $\max_{0\leq\beta\leq 1}\sum_{t=1}^{\frac{T}{2}}4\alpha_t\beta-2\beta+1-2\alpha_t\leq o(T)$ which also implies that $\sum_{t=1}^{\frac{T}{2}}2\alpha_t-1\leq o(T)$ and $\sum_{t=1}^{\frac{T}{2}}1-2\alpha_t\leq o(T)$ since $\sum_{t=1}^{\frac{T}{2}}4\alpha_t\beta-2\beta+1-2\alpha_t$ is a linear function of β and thus its maximum occurs at $\beta=0$ or $\beta=1$.

For scenario 2 $\max_{y\in\Delta_Y}\sum_{t=1}^{T}x_t^\top A_t y-\min_{x\in X}\max_{y\in Y}\sum_{t=1}^{T}x^\top A_t y\leq o(T)$ reduces to

$\max_{0\leq\beta\leq 1}\sum_{t=1}^{\frac{T}{2}}4\alpha_t\beta-2\beta+1-2\alpha_t+\frac{T}{2}(2\beta-1)\leq o(T)$ which implies $\sum_{t=1}^{\frac{T}{2}}2\alpha_t-1+\frac{T}{2}\leq o(T)$ and $\sum_{t=1}^{\frac{T}{2}}1-2\alpha_t+\frac{T}{2}\leq o(T)$. Finally, notice that $\sum_{t=1}^{\frac{T}{2}}2\alpha_t-1+\frac{T}{2}\leq o(T)$ implies $\frac{T}{2}\leq o(T)+\sum_{t=1}^{\frac{T}{2}}1-2\alpha_t$ but from scenario 1 we have that $\sum_{t=1}^{\frac{T}{2}}1-2\alpha_t\leq o(T)$ since $\frac{T}{2}\leq o(T)$ is a contradiction we get the result. □

4.4 Online Matrix Games: Full Information

4.4.1 Saddle Point Regularized Follow-the-Leader

In this section we propose an algorithm to solve the OMG problem in the full information setting. In fact, we will consider the algorithm in a slighly more general setting than the OMG problem, allowing the sequence of payoff functions to be specified by arbitrary

convex-concave Lipschitz functions, and the action sets of Player 1 and Player 2 ($X \subset \mathbb{R}^n$ and $Y \subset \mathbb{R}^n$ respectively) to be arbitrary convex compact sets.

Let the sequence of convex-concave functions be $\{\bar{\mathcal{L}}_t(x,y)\}_{t=1}^T$, which are $G_{\bar{\mathcal{L}}}$-Lipschitz with respect to some norm $\|\cdot\|$. We propose an algorithm called Saddle Point Regularized Follow the Leader (SP-RFTL), shown in Algorithm 9.

Algorithm 9 Saddle-Point Regularized-Follow-the-Leader (SP-RFTL)

input: $x_1 \in X$, $y_1 \in Y$, parameters: $\eta > 0$, strongly convex functions R_X, R_Y
for $t = 1, ...T$ **do**
  Play (x_t, y_t)
  Observe $\bar{\mathcal{L}}_t$
  $\mathcal{L}_t(x,y) \leftarrow \bar{\mathcal{L}}_t + \frac{1}{\eta}R_X(x) - \frac{1}{\eta}R_Y(y)$
  $x_{t+1} \leftarrow \arg\min_{x\in X} \max_{y\in Y} \sum_{\tau=1}^t \mathcal{L}_t(x,y)$
  $y_{t+1} \leftarrow \arg\max_{y\in Y} \min_{x\in X} \sum_{\tau=1}^t \mathcal{L}_t(x,y)$
end for

The regularizers R_X, R_Y are used as input for the algorithm. We will choose regularizers that are strongly convex with respect to norm $\|\cdot\|$, and G_{R_1} and G_{R_2} Lipschitz with respect to norm $\|\cdot\|$, which means that $\|\nabla R_X(x)\|_* \leq G_{R_1}$ for all $x \in X$, and $\|\nabla R_Y(y)\|_* \leq G_{R_2}$ for, all $y \in Y$. Finally, we assume $R_X(x) \geq 0$ for all $x \in X$ and $R_Y(y) \geq 0$ for all $y \in Y$.

The main difference between SP-RFTL and the well known Regularized Follow the Leader (RFTL) algorithm [6, 116] is that in SP-RFTL both players update jointly and play the saddle point of the sum of regularized games observed so far. In particular, they disregard their previous actions. In contrast, the updates for RFTL would be

$$x_{t+1}^{RFTL} \leftarrow \arg\min_{x\in X} \sum_{\tau=1}^t \left[\bar{\mathcal{L}}_\tau(x, y_\tau^{RFTL}) + \frac{1}{\eta}R_X(x)\right]$$
$$y_{t+1}^{RFTL} \leftarrow \arg\max_{y\in Y} \sum_{\tau=1}^t \left[\bar{\mathcal{L}}_\tau(x_\tau^{RFTL}, y) - \frac{1}{\eta}R_Y(y)\right]$$

for $t = 2, ..., T$, and x_1^{RFTL}, y_1^{RFTL} are chosen as to minimize $R_X(x)$ and $-R_Y(y)$ in their respective sets X, Y. It is easy to see that the sequence of iterates is in general not the same.

In fact, in view of Theorem 21 we know that RFTL can not achieve sublinear NE regret when the sequence of functions is chosen arbitrarily.

We have the following guarantee for SP-RFTL.

Theorem 22. *For $t = 1, ..., T$, let $\bar{\mathcal{L}}_t$ be $G_{\bar{\mathcal{L}}}$-Lipschitz with respect to norm $\|\cdot\|$. Let R_X, R_Y be strongly convex functions with respect to the same norm, let G_{R_X}, G_{R_Y} be the Lipschitz constants of R_X, R_Y with respect to the same norm. Let $\{(x_t, y_t)\}_{t=1}^T$ be the iterates generated by* SP-RFTL *when run on convex-concave functions $\{\bar{\mathcal{L}}_t(x, y)\}_{t=1}^T$. It holds that*

$$\begin{aligned}
&\left|\sum_{t=1}^{T} \bar{\mathcal{L}}_t(x_t, y_t) - \min_{x\in X}\max_{y\in Y}\sum_{t=1}^{T} \bar{\mathcal{L}}_t(x, y)\right| \\
\leq& 8\eta\left[G_{\bar{\mathcal{L}}} + \frac{1}{\eta}\max(G_{R_X}, G_{R_Y})\right]^2 (1 + \ln(T)) \\
&+ \frac{T}{\eta}\max_{y\in Y} R_Y(y) + \frac{T}{\eta}\max_{x\in X} R_X(x) \;=\; O\left(\sqrt{T\ln(T)}\right),
\end{aligned}$$

where the last equality follows by choosing $\eta = \frac{\sqrt{T}}{\ln(T)}$.

A formal proof of the theorem will be given shortly.

We note that the bound in Theorem 22 holds for general convex-concave functions, however the dependence on the dimension is hidden on the Lipschitz constants and the choice of regularizer. It is easy to check that if one chooses $\|\cdot\|_2^2$ as regularizer, and the functions $\{\mathcal{L}_t\}_{t=1}^T$ are G-Lipschitz continuous with respect to norm $\|\cdot\|_2^2$, then the NE regret bound will be $O(n\ln(T)\sqrt{T})$.

We now provide a sketch of the proof of Theorem 22. Define $\mathcal{L}_t(x, y) \triangleq \bar{\mathcal{L}}_t(x, y) + \frac{1}{\eta}R_X(x) - \frac{1}{\eta}R_Y(y)$. Notice that it is $\frac{1}{\eta}$-strongly convex in x with respect to norm $\|\cdot\|$ for all $y \in Y$ and $\frac{1}{\eta}$-strongly concave with respect to norm $\|\cdot\|$ for all $x \in X$. Additionally, notice that $\mathcal{L}_t$ is $G_{\mathcal{L}} \triangleq G_{\bar{\mathcal{L}}} + \frac{1}{\eta}(G_{R_X} + G_{R_Y})$-Lipschitz with respect to norm $\|\cdot\|$. Finally,

notice that for $t = 1, ..., T$, all $x \in X$ and all $y \in Y$ it holds that

$$-\frac{1}{\eta}R_Y(y) \leq \mathcal{L}_t(x, y) - \bar{\mathcal{L}}_t(x, y) \leq \frac{1}{\eta}R_X(x) \tag{4.6}$$

The following lemma shows that the value of the convex-concave games defined by $\sum_{t=1}^{T} \mathcal{L}_t$ and $\sum_{t=1}^{T} \bar{\mathcal{L}}_t$ are not too far from each other.

Lemma 34. *Let*

$$\bar{x}_{T+1} \in \arg\min_{x \in X} \max_{y \in Y} \sum_{t=1}^{T} \bar{\mathcal{L}}_t(x, y),$$
$$\bar{y}_{T+1} \in \arg\max_{y \in Y} \min_{x \in X} \sum_{t=1}^{T} \bar{\mathcal{L}}_t(x, y).$$

It holds that

$$\begin{aligned}
&-\frac{T}{\eta}R_Y(\bar{y}_{T+1}) \\
&\leq \min_{x \in X} \max_{y \in Y} \sum_{t=1}^{T} \mathcal{L}_t(x, y) - \min_{x \in X} \max_{y \in Y} \sum_{t=1}^{T} \bar{\mathcal{L}}_t(x, y) \\
&\leq \frac{T}{\eta}R_X(\bar{x}_{T+1}).
\end{aligned}$$

Proof of Lemma 34.

$$\begin{aligned}
\min_{x \in X} \max_{y \in Y} \sum_{t=1}^{T} \mathcal{L}_t(x, y) &= \sum_{t=1}^{T} [\bar{\mathcal{L}}_t(x_{T+1}, y_{T+1}) + \frac{1}{\eta}R_X(x_{T+1}) - \frac{1}{\eta}R_Y(y_{T+1})] \\
&\leq \sum_{t=1}^{T} [\bar{\mathcal{L}}_t(\bar{x}_{T+1}, y_{T+1}) + \frac{1}{\eta}R_X(\bar{x}_{T+1}) - \frac{1}{\eta}R_Y(y_{T+1})] \\
&\quad \text{by Equation (4.2)} \\
&\leq \sum_{t=1}^{T} [\bar{\mathcal{L}}_t(\bar{x}_{T+1}, \bar{y}_{T+1}) + \frac{1}{\eta}R_X(\bar{x}_{T+1}) - \frac{1}{\eta}R_Y(y_{T+1})] \\
&\quad \text{by Equation (4.2)}
\end{aligned}$$

$$= \min_{x\in X}\max_{y\in Y}\sum_{t=1}^{T}[\bar{\mathcal{L}}_t(x,y) + \frac{T}{\eta}R_X(\bar{x}_{T+1}) - \frac{T}{\eta}R_Y(y_{T+1})]$$
$$\leq \min_{x\in X}\max_{y\in Y}\sum_{t=1}^{T}\bar{\mathcal{L}}_t(x,y) + \frac{T}{\eta}R_X(\bar{x}_{T+1}).$$

The other inequality can be obtained by a similar argument. □

To prove the NE regret bound, we note that SP-RFTL is running a Follow-the-Leader scheme on functions $\{\mathcal{L}_{t=1}^{T}\}$ [82]. With the next two lemmas one can show that the NE regret of the players relative to functions $\{\mathcal{L}\}_{t=1}^{T}$ is small.

Lemma 35. *Let* $\{(x_t, y_t)\}_{t=1}^{T}$ *be the iterates of* SP-RFTL. *It holds that*

$$\begin{aligned}
&- G_{\mathcal{L}}\textstyle\sum_{t=1}^{T}\|x_t - x_{t+1}\| \\
&\leq \sum_{t=1}^{T}\mathcal{L}_t(x_{t+1}, y_{t+1}) - \min_{x\in X}\max_{y\in Y}\sum_{t=1}^{T}\mathcal{L}_t(x,y) \\
&\leq G_{\mathcal{L}}\textstyle\sum_{t=1}^{T}\|y_t - y_{t+1}\|.
\end{aligned}$$

Proof of Lemma 35. We first prove the second inequality. We proceed by induction. The base case $t = 1$ holds by definition of (x_2, y_2), indeed

$$\mathcal{L}_1(x_2, y_2) + G_{\mathcal{L}}\|y_1 - y_2\| \geq \mathcal{L}_1(x_2, y_2) := \min_{x\in X}\max_{y\in Y}\mathcal{L}_1(x,y).$$

We now assume the following claim holds for $T-1$:

$$\min_{x\in X}\max_{y\in Y}\sum_{t=1}^{T-1}\mathcal{L}_t(x,y) \geq \sum_{t=1}^{T-1}\mathcal{L}_t(x_{t+1}, y_{t+1}) - G_{\mathcal{L}}\sum_{t=1}^{T-1}\|y_t - y_{t+1}\|, \tag{4.7}$$

and show it holds for T.

$$\min_{x\in X}\max_{y\in Y}\sum_{t=1}^{T}\mathcal{L}_t(x,y)$$

$$
\begin{aligned}
&= \sum_{t=1}^{T-1} \mathcal{L}_t(x_{T+1}, y_{T+1}) + \mathcal{L}_T(x_{T+1}, y_{T+1}) \\
&\geq \sum_{t=1}^{T-1} \mathcal{L}_t(x_{T+1}, y_T) + \mathcal{L}_T(x_{T+1}, y_T) && \text{by Equation (4.2)} \\
&\geq \sum_{t=1}^{T-1} \mathcal{L}_t(x_T, y_T) + \mathcal{L}_T(x_{T+1}, y_T) && \text{by Equation (4.2)} \\
&\geq \sum_{t=1}^{T-1} \mathcal{L}_t(x_{t+1}, y_{t+1}) - G_{\mathcal{L}} \sum_{t=1}^{T-1} \|y_t - y_{t+1}\| + \mathcal{L}_T(x_{T+1}, y_T) && \text{by Equation (4.12)} \\
&= \sum_{t=1}^{T} \mathcal{L}_t(x_{t+1}, y_{t+1}) - G_{\mathcal{L}} \sum_{t=1}^{T-1} \|y_t - y_{t+1}\| + \mathcal{L}_T(x_{T+1}, y_T) - \mathcal{L}_T(x_{T+1}, y_{T+1}) \\
&\geq \sum_{t=1}^{T} \mathcal{L}_t(x_{t+1}, y_{t+1}) - G_{\mathcal{L}} \sum_{t=1}^{T-1} \|y_t - y_{t+1}\| - G_{\mathcal{L}} \|y_T - y_{T+1}\| \\
&= \sum_{t=1}^{T} \mathcal{L}_t(x_{t+1}, y_{t+1}) - G_{\mathcal{L}} \sum_{t=1}^{T} \|y_t - y_{t+1}\|.
\end{aligned}
$$

We now show by induction that

$$
\min_{x \in X} \max_{y \in Y} \sum_{t=1}^{T} \mathcal{L}_t(x, y) \leq \sum_{t=1}^{T} \mathcal{L}_t(x_{t+1}, y_{t+1}) + G_{\mathcal{L}} \sum_{t=1}^{T} \|x_t - x_{t+1}\|.
$$

Indeed, $t = 1$ follows from the definition of (x_2, y_2). We now assume the claim holds for $T - 1$ and prove it for T:

$$
\begin{aligned}
&\min_{x \in X} \max_{y \in Y} \sum_{t=1}^{T} \mathcal{L}_t(x, y) \\
&= \sum_{t=1}^{T} \mathcal{L}_t(x_{T+1}, y_{T+1}) \\
&\leq \sum_{t=1}^{T-1} \mathcal{L}_t(x_T, y_{T+1}) + \mathcal{L}_T(x_T, y_{T+1}) && \text{by Equation (4.2)} \\
&\leq \sum_{t=1}^{T-1} \mathcal{L}_t(x_T, y_T) + \mathcal{L}_T(x_T, y_{T+1}) && \text{by Equation (4.2)} \\
&\leq \sum_{t=1}^{T-1} \mathcal{L}_t(x_{t+1}, y_{t+1}) + G_{\mathcal{L}} \sum_{t=1}^{T-1} \|x_t - x_{t+1}\|
\end{aligned}
$$

$$+ \mathcal{L}_T(x_T, y_{T+1}) + \mathcal{L}_T(x_{T+1}, y_{T+1}) - \mathcal{L}_T(x_{T+1}, y_{T+1}) \qquad \text{by induction claim}$$

$$\leq \sum_{t=1}^{T} \mathcal{L}_t(x_{t+1}, y_{t+1}) + G_{\mathcal{L}} \sum_{t=1}^{T} \|x_t - x_{t+1}\| \qquad \text{since } \mathcal{L}_T \text{ is } G_{\mathcal{L}}\text{-Lipschitz.}$$

□

Lemma 36. *Let $\{(x_t, y_t)\}_{t=1}^T$ be the sequence of iterates generated by the algorithm. It holds that*

$$\begin{aligned} &\|x_t - x_{t+1}\| + \|y_t - y_{t+1}\| \\ &\leq \frac{4\eta}{t} \left[G_{\bar{\mathcal{L}}} + \frac{1}{\eta} \max(G_{R_X}, G_{R_Y}) \right]. \end{aligned}$$

Proof of Lemma 36. Fix t, define $J(x, y) \triangleq \sum_{\tau=1}^{t-1} \mathcal{L}_\tau(x, y) + \mathcal{L}_t(x, y)$ and notice it is $\frac{t}{\eta}$-strongly convex strongly concave with respect to norm $\|\cdot\|$. Also notice that (x_{t+1}, y_{t+1}) is the unique saddle point of $J(x, y)$.

By strong convexity of J and definition of x_{t+1} it holds that for any $x \in X$ and any $y \in Y$

$$J(x, y) \geq J(x_{t+1}, y) + \nabla_x J(x_{t+1}, y)^\top (x - x_{t+1}) + \frac{t}{2\eta} \|x - x_{t+1}\|^2.$$

Plugging in $y = y_{t+1}$ and recalling the KKT condition $\nabla_x J(x_{t+1}, y_{t+1})^\top (x - x_{t+1}) \geq 0$, we have that for any $x \in X$

$$\frac{2\eta}{t} \big[J(x, y_{t+1}) - J(x_{t+1}, y_{t+1}) \big] \geq \|x - x_{t+1}\|^2. \tag{4.8}$$

Similarly, since J is $\frac{t}{\eta}$ strongly concave. That is, for any $y \in Y$

$$J(x_{t+1}, y) \leq J(x_{t+1}, y_{t+1}) + \nabla_y J(x_{t+1}, y_{t+1})^\top (y - y_{t+1}) - \frac{t}{2\eta} \|y - y_{t+1}\|^2.$$

Together with the KKT condition $\nabla_y J(x_{t+1}, y_{t+1})^\top (y - y_{t+1}) \leq 0$ we get that for any

$y \in Y$

$$\frac{2\eta}{t}\big[J(x_{t+1}, y_{t+1}) - J(x_{t+1}, y)\big] \geq \|y - y_{t+1}\|^2. \tag{4.9}$$

Adding up Equations (4.13) and (4.14), plugging $x = x_t$ and $y = y_t$ we get

$$\begin{aligned}
&\frac{2\eta}{t}\big[J(x_t, y_{t+1}) - J(x_{t+1}, y_t)\big] \geq \|x_t - x_{t+1}\|^2 + \|y - y_{t+1}\|^2. \\
\iff &\frac{2\eta}{t}\big[\sum_{\tau=1}^{t-1}\mathcal{L}_\tau(x_t, y_{t+1}) + \mathcal{L}_t(x_t, y_{t+1}) - [\sum_{\tau=1}^{t-1}\mathcal{L}_\tau(x_{t+1}, y_t) + \mathcal{L}_t(x_{t+1}, y_t)]\big] \\
&\qquad \geq \|x_t - x_{t+1}\|^2 + \|y - y_{t+1}\|^2 \\
\implies &\frac{2\eta}{t}\big[\sum_{\tau=1}^{t-1}\mathcal{L}_\tau(x_t, y_t) + \mathcal{L}_t(x_t, y_{t+1}) - [\sum_{\tau=1}^{t-1}\mathcal{L}_\tau(x_{t+1}, y_t) + \mathcal{L}_t(x_{t+1}, y_t)]\big] \\
&\qquad \geq \|x_t - x_{t+1}\|^2 + \|y - y_{t+1}\|^2,
\end{aligned}$$

since $\sum_{\tau=1}^{t-1}\mathcal{L}_\tau(x_t, y_{t+1}) \leq \sum_{\tau=1}^{t-1}\mathcal{L}_\tau(x_t, y_t)$.

Additionally, since $\sum_{\tau=1}^{t-1}\mathcal{L}_\tau(x_t, y_t) \leq \sum_{\tau=1}^{t-1}\mathcal{L}_\tau(x_{t+1}, y_t)$, we have

$$\begin{aligned}
&\frac{2\eta}{t}\big[\sum_{\tau=1}^{t-1}\mathcal{L}_\tau(x_t, y_t) + \mathcal{L}_t(x_t, y_{t+1}) - \sum_{\tau=1}^{t-1}\mathcal{L}_\tau(x_t, y_t) - \mathcal{L}_t(x_{t+1}, y_t)\big] \\
&\qquad \geq \|x_t - x_{t+1}\|^2 + \|y - y_{t+1}\|^2 \\
\iff &\frac{2\eta}{t}\big[\mathcal{L}_t(x_t, y_{t+1}) - \mathcal{L}_t(x_{t+1}, y_t)\big] \geq \|x_t - x_{t+1}\|^2 + \|y_t - y_{t+1}\|^2 \\
\iff &\frac{2\eta}{t}\big[\bar{\mathcal{L}}_t(x_t, y_{t+1}) + \frac{1}{\eta}R_X(x_t) - \frac{1}{\eta}R_Y(y_{t+1}) - \bar{\mathcal{L}}_t(x_{t+1}, y_t) - \frac{1}{\eta}R_X(x_{t+1}) + \frac{1}{\eta}R_Y(y_t)\big] \\
&\qquad \geq \|x_t - x_{t+1}\|^2 + \|y_t - y_{t+1}\|^2 \\
\implies &\frac{2\eta}{t}\big[G_{\bar{\mathcal{L}}}\|[x_t; y_{t+1}] - [x_{t+1}; y_t]\| + \frac{1}{\eta}R_X(x_t) - \frac{1}{\eta}R_X(x_{t+1}) + \frac{1}{\eta}R_Y(y_t) - \frac{1}{\eta}R_Y(y_{t+1})\big] \\
&\qquad \geq \|x_t - x_{t+1}\|^2 + \|y_t - y_{t+1}\|^2 \\
\implies &\frac{2\eta}{t}\big[G_{\bar{\mathcal{L}}}\|x_t - x_{t+1}\| + G_{\bar{\mathcal{L}}}\|y_t - y_{t+1}\| + \frac{1}{\eta}R_X(x_t) - \frac{1}{\eta}R_X(x_{t+1}) \\
&\qquad + \frac{1}{\eta}R_Y(y_t) - \frac{1}{\eta}R_Y(y_{t+1})\big] \geq \|x_t - x_{t+1}\|^2 + \|y_t - y_{t+1}\|^2 \\
\implies &\frac{2\eta}{t}\big[G_{\bar{\mathcal{L}}}\|x_t - x_{t+1}\| + G_{\bar{\mathcal{L}}}\|y_t - y_{t+1}\| + \frac{G_{R_X}}{\eta}\|x_t - x_{t+1}\| + \frac{G_{R_Y}}{\eta}\|y_t - y_{t+1}\|\big]
\end{aligned}$$

$$
\begin{aligned}
&\geq \|x_t - x_{t+1}\|^2 + \|y_t - y_{t+1}\|^2 \\
\implies &\frac{2\eta}{t}[G_{\bar{\mathcal{L}}} + \frac{1}{\eta}\max(G_{R_X}, G_{R_Y})]\big[\|x_t - x_{t+1}\| + \|y_t - y_{t+1}\|\big] \\
&\geq \|x_t - x_{t+1}\|^2 + \|y_t - y_{t+1}\|^2 \\
\iff &\frac{2\eta}{t}[G_{\bar{\mathcal{L}}} + \frac{1}{\eta}\max(G_{R_X}, G_{R_Y})] \geq \frac{\|x_t - x_{t+1}\|^2 + \|y_t - y_{t+1}\|^2}{\|x_t - x_{t+1}\| + \|y_t - y_{t+1}\|}.
\end{aligned}
$$

Finally, since x^2 is a convex function, by Jensen's inequality: $\frac{a^2}{2} + \frac{b^2}{2} \geq \left(\frac{a+b}{2}\right)^2$, we have $a^2 + b^2 \geq \frac{(a+b)^2}{2}$. This, together with the last implication, yields the result

$$
\frac{4\eta}{t}[G_{\bar{\mathcal{L}}} + \frac{1}{\eta}\max(G_{R_X}, G_{R_Y})] \geq \|x_t - x_{t+1}\| + \|y_t - y_{t+1}\|.
$$

□

Combining the NE regret bound obtained on functions $\{\mathcal{L}\}_{t=1}^T$ together with Lemma 34 and equation (4.6) will yield the theorem as shown next.

Proof of Theorem 22.

$$
\begin{aligned}
&\sum_{t=1}^T \bar{\mathcal{L}}_t(x_t, y_t) - \min_{x\in X}\max_{y\in Y}\sum_{t=1}^T \bar{\mathcal{L}}_t(x, y) \\
&\leq \sum_{t=1}^T \mathcal{L}_t(x_t, y_t) - \min_{x\in X}\max_{y\in Y}\sum_{t=1}^T \bar{\mathcal{L}}_t(x, y) + \sum_{t=1}^T \frac{1}{\eta}R_Y(y_t) \quad \text{by Equation 4.6} \\
&\leq \sum_{t=1}^T \mathcal{L}_t(x_t, y_t) - \min_{x\in X}\max_{y\in Y}\sum_{t=1}^T \mathcal{L}_t(x, y) + \sum_{t=1}^T \frac{1}{\eta}R_Y(y_t) + \frac{T}{\eta}R_X(x_{T+1}) \quad \text{by Lemma 34} \\
&\leq \sum_{t=1}^T \mathcal{L}_t(x_t, y_t) - \sum_{t=1}^T \mathcal{L}_t(x_{t+1}, y_{t+1}) + \sum_{t=1}^T \frac{1}{\eta}R_Y(y_t) + \frac{T}{\eta}R_X(x_{T+1}) \\
&\qquad + G_{\mathcal{L}}\sum_{t=1}^T \|y_t - y_{t+1}\| \quad \text{by Lemma 35} \\
&\leq \sum_{t=1}^T G_{\mathcal{L}}(\|x_t - x_{t+1}\| + \|y_t - y_{t+1}\|) + \sum_{t=1}^T \frac{1}{\eta}R_Y(y_t) + \frac{T}{\eta}R_X(x_{T+1})
\end{aligned}
$$

$$
\begin{aligned}
&\quad + G_{\mathcal{L}} \sum_{t=1}^{T} \|y_t - y_{t+1}\| \\
&\leq \sum_{t=1}^{T} G_{\mathcal{L}}(\|x_t - x_{t+1}\| + \|y_t - y_{t+1}\|) + \sum_{t=1}^{T} \frac{1}{\eta} R_Y(y_t) + \frac{T}{\eta} R_X(x_{T+1}) \\
&+ G_{\mathcal{L}} \sum_{t=1}^{T} \|y_t - y_{t+1}\| \\
&\leq 2 \sum_{t=1}^{T} G_{\mathcal{L}}(\|x_t - x_{t+1}\| + \|y_t - y_{t+1}\|) + \sum_{t=1}^{T} \frac{1}{\eta} R_Y(y_t) + \frac{T}{\eta} R_X(x_{T+1}) \\
&\leq 2 \sum_{t=1}^{T} G_{\mathcal{L}}(\frac{4\eta}{t}[G_{\bar{\mathcal{L}}} + \frac{1}{\eta} \max(G_{R_X}, G_{R_Y})]) + \sum_{t=1}^{T} \frac{1}{\eta} R_Y(y_t) + \frac{T}{\eta} R_X(x_{T+1}) \\
&\leq 8 G_{\mathcal{L}} \eta [G_{\bar{\mathcal{L}}} + \frac{1}{\eta} \max(G_{R_X}, G_{R_Y})](1 + \int_1^T \frac{1}{t} dt) + \sum_{t=1}^{T} \frac{1}{\eta} R_Y(y_t) + \frac{T}{\eta} R_X(x_{T+1}) \\
&\leq 8 G_{\mathcal{L}} \eta [G_{\bar{\mathcal{L}}} + \frac{1}{\eta} \max(G_{R_X}, G_{R_Y})](1 + \ln(T)) + \frac{T}{\eta} \max_{y \in Y} R_Y(y) + \frac{T}{\eta} \max_{x \in X} R_X(x) \\
&\leq 8 \eta [G_{\bar{\mathcal{L}}} + \frac{1}{\eta} \max(G_{R_X}, G_{R_Y})]^2 (1 + \ln(T)) + \frac{T}{\eta} \max_{y \in Y} R_Y(y) + \frac{T}{\eta} \max_{x \in X} R_X(x).
\end{aligned}
$$

Notice that $\min_{x \in X} \max_{y \in Y} \sum_{t=1}^{T} \bar{\mathcal{L}}_t(x, y) - \sum_{t=1}^{T} \bar{\mathcal{L}}_t(x_t, y_t)$ can be upper bounded by the same quantity using the same argument. This concludes the proof. □

4.4.2 Logarithmic Dependence on the Dimension of the Action Spaces

Previously, we analyzed the OMG problem by treating the payoff functions as general convex-concave functions and the action spaces as general convex compact sets. We explained that in general one should expect to achieve NE regret which depends linearly in the dimension of the problem. The goal in this section is to obtain sharper NE regret bounds that scale as $O(\ln(T)\sqrt{T}\ln(\max(d_1, d_2)))$ by exploiting the geometry of the decision sets Δ_X, Δ_Y and the bilinear structure of the payoff functions. This allows us to solve games which may have exponentially many actions, which often arise in combinatorial optimization settings.

The plan to obtain the desired NE regret bounds in this more restrictive setting is to use the negative entropy as a regularization function (which is strongly convex with respect to

$\|\cdot\|_1$), that is $R_X(x) = \sum_{i=1}^{d_1} x_i \ln(x_i) + \ln(d_1)$ and $R_Y(y) = \sum_{i=1}^{d_2} y_i \ln(y_i) + \ln(d_2)$ where the extra logarithmic terms ensure R_X, R_Y are nonnegative everywhere in their respective simplexes. Unfortunately, the negative entropy is not Lipschitz over the simplex, so we can not leverage our result from Theorem 22. To deal with this challenge, we will restrict the new algorithm to play over a restricted simplex:[3]

$$\Delta_\theta = \{z \in \mathbb{R}^d : \|z\|_1 = 1, z_i \geq \theta, i = 1, ..., d\}. \tag{4.10}$$

The tuning parameter $\theta \in [0, 1/d]$ used for the algorithm will be defined later in the analysis. (Notice that when $\theta > 1/d$, the set is empty.) We have the following result.

Lemma 37. *The function $R(x) \triangleq \sum_{i=1}^{d} x_i \ln(x_i)$ is G_R-Lipschitz continuous with respect to $\|\cdot\|_1$ over Δ_θ with $G_R = \max\{|\ln(\theta)|, 1\}$.*

Proof of Lemma 37. We need to find $G_R > 0$ such that $\|\nabla R(x)\|_\infty \leq G_R$ for all $x \in \Delta_\theta$. Notice that $[\nabla R(x)]_i = 1 + \ln(x_i)$ for $i = 1, ...d$. Moreover, since for every $i = 1, ..., d$ we have $\theta \leq x_i \leq 1$ the following sequence of inequalities hold: $\ln(\theta) \leq 1 + \ln(\theta) \leq 1 + \ln(x_i) \leq 1$. It follows that $G_R = \max\{|\ln(\theta)|, 1\}$. □

The algorithm ONLINE-MATRIX-GAMES REGULARIZED-FOLLOW-THE-LEADER is an instantiation of SP-RFTL with a particular choice of regularization functions, which are nonegative and Lipschitz over the sets $\Delta_{X,\theta}$, $\Delta_{Y,\theta}$. With this, we can prove a NE regret bound for the OMG problem. For the remainder of the chapter, the regularization functions will be set as follows:

$$\begin{aligned} R_X(x) &\triangleq \sum_{i=1}^{d_1} x_i \ln(x_i) + \ln(d_1), \\ R_Y(y) &\triangleq \sum_{i=1}^{d_2} y_i \ln(y_i) + \ln(d_2). \end{aligned}$$

[3] We will also use the notation $\Delta_{X,\theta}$ and $\Delta_{Y,\theta}$ to mean the restricted simplex of Player 1 and 2, respectively

Algorithm 10 Online-Matrix-Games Regularized-Follow-the-Regularized-Leader (OMG-RFTL)

input: $x_1 \in \Delta_{X,\theta} \subset \mathbb{R}^{d_1}$, $y_1 \in \Delta_{Y,\theta} \subset \mathbb{R}^{d_2}$, parameters: $\eta > 0$, $\theta < \min\{\frac{1}{d_1}, \frac{1}{d_2}\}$.
for $t = 1, ...T$ **do**
  Play (x_t, y_t), observe matrix A_t
  $\bar{\mathcal{L}}_t \leftarrow x^\top A_t y$
  $\mathcal{L}_t(x, y) \leftarrow \bar{\mathcal{L}}_t + \frac{1}{\eta} R_X(x) - \frac{1}{\eta} R_Y(y)$
  $x_{t+1} \leftarrow \arg\min_{x \in \Delta_{X,\theta}} \max_{y \in \Delta_{Y,\theta}} \sum_{\tau=1}^{t} \mathcal{L}_t(x, y)$
  $y_{t+1} \leftarrow \arg\max_{y \in \Delta_{Y,\theta}} \min_{x \in \Delta_{X,\theta}} \sum_{\tau=1}^{t} \mathcal{L}_t(x, y)$
end for

We have the following guarantee for OMG-RFTL.

Theorem 23. *Let $\{A_t\}_{t=1}^T$ be an arbitrary sequence of matrices with entries bounded between $[-1, 1]$. Let $G_{\bar{\mathcal{L}}}$ be the Lipschitz constant (with respect to $\|\cdot\|_1$) of $\bar{\mathcal{L}}_t \triangleq x^\top A_t y$ for $t = 1, ..., T$. Let $\{(x_t, y_t)\}_{t=1}^T$ be the iterates of* OMG-RFTL*) and choose $\theta = e^{-\eta G_{\bar{\mathcal{L}}}} \leq \min\{\frac{1}{d_1}, \frac{1}{d_2}\}$ such that $\frac{|\ln(\theta)|}{\eta} = G_{\bar{\mathcal{L}}}$. Set $\eta = \frac{\sqrt{T}}{G_{\bar{\mathcal{L}}}}$. It holds that*

$$\begin{aligned}
&\left| \sum_{t=1}^{T} x_t^\top A_t y_t - \min_{x \in \Delta} \max_{y \in \Delta} \sum_{t=1}^{T} x^\top A_t y \right| \\
&\leq 32 G_{\bar{\mathcal{L}}} \sqrt{T}(1 + \ln(T)) + 2\sqrt{T} \max\{\ln d_1, \ln d_2\} + \\
&\quad 2 \max\{d_1, d_2\} G_{\bar{\mathcal{L}}} T e^{-\sqrt{T}} \\
&= O\left(\ln(T)\sqrt{T} + \sqrt{T} \max\{\ln d_1, \ln d_2\}\right) + \\
&\quad o(1) \max\{d_1, d_2\}.
\end{aligned}$$

A full proof of the theorem will be given shortly. We now give a sketch of the proof. Since the algorithm selects actions over the restricted simplex, we must quantify the potential loss in the NE regret bound imposed by this restriction. The next two lemmas make this precise.

Lemma 38. *Let $z^* \in \Delta \subset \mathbb{R}^d$ define $z_p^* \triangleq \arg\min_{z \in \Delta_\theta} \|z - z^*\|_1$, with $\theta \leq \frac{1}{d}$. Notice z_p^* is unique since it is a projection. It holds that $\|z_p^* - z^*\|_1 \leq 2\theta(d - 1)$.*

Proof of Lemma 38. Choose $z^* = [1; 0; 0; ...; 0; 0]$, it is easy to see that $z_p^* = [1 - \theta(d -$

$1); \theta; \theta; ...; \theta, \theta]$ and $\|z^* - z_p^*\|_1 = 2\theta(d-1)$. $\square$

Lemma 39. *Let $\{\bar{\mathcal{L}}_t(x,y)\}_{t=1}^T$ be an arbitrary sequence of convex-concave functions, $\bar{\mathcal{L}}_t : \Delta_X \times \Delta_Y \to \mathbb{R}$, that are $G_{\bar{\mathcal{L}}}$-Lipschitz with respect to $\|\cdot\|_1$. With $\Delta_X \subseteq \mathbb{R}^{d_1}$, and $\Delta_Y \subseteq \mathbb{R}^{d_2}$. It holds that*

$$\begin{aligned}
&- G_{\bar{\mathcal{L}}} T \|x_p^* - x^*\|_1 \\
\leq &\min_{x\in\Delta}\max_{y\in\Delta} \sum_{t=1}^T \bar{\mathcal{L}}_t(x,y) - \min_{x\in\Delta_\theta}\max_{y\in\Delta_\theta} \sum_{t=1}^T \bar{\mathcal{L}}_t(x,y) \\
\leq &\, G_{\bar{\mathcal{L}}} T \|y_p^* - y^*\|_1.
\end{aligned}$$

Proof of Lemma 39. Let (x^*, y^*) be any saddle point pair for $\sum_{t=1}^T \bar{\mathcal{L}}_t(x,y)$ with $x^* \in \Delta, y^* \in \Delta$. Let (x_θ^*, y_θ^*) be any saddle point pair for $\sum_{t=1}^T \bar{\mathcal{L}}_t(x,y)$ with $x_\theta^* \in \Delta, y_\theta^* \in \Delta$. Let x_p^*, y_p^* be the projection of x^*, y^* onto the respective simplexes using the $\|\cdot\|_\infty$ norm. We first show the second inequality. Notice that

$$\begin{aligned}
\sum_{t=1}^T \bar{\mathcal{L}}_t(x^*, y^*) &\leq \sum_{t=1}^T \bar{\mathcal{L}}_t(x_\theta^*, y^*) \\
&\leq \sum_{t=1}^T \bar{\mathcal{L}}_t(x_\theta^*, y_p^*) + G_{\bar{\mathcal{L}}} T \|y_p^* - y^*\|_1 \\
&\leq \sum_{t=1}^T \bar{\mathcal{L}}_t(x_\theta^*, y_\theta^*) + G_{\bar{\mathcal{L}}} T \|y_p^* - y^*\|_1.
\end{aligned}$$

To show the first inequality in the statement of the lemma notice that

$$\begin{aligned}
\sum_{t=1}^T \bar{\mathcal{L}}_t(x^*, y^*) &\geq \sum_{t=1}^T \bar{\mathcal{L}}_t(x^*, y_\theta^*) \\
&\geq \sum_{t=1}^T \bar{\mathcal{L}}_t(x_p^*, y_\theta^*) - G_{\bar{\mathcal{L}}} T \|x_p^* - x^*\|_1 \\
&\geq \sum_{t=1}^T \bar{\mathcal{L}}_t(x_\theta^*, y_\theta^*) - G_{\bar{\mathcal{L}}} T \|x_p^* - x^*\|_1.
\end{aligned}$$

This concludes the proof. □

Combining the previous two lemmas and Theorem 22, one can show the NE regret bound for OMG-RFTL holds.

We are ready to proof Theorem 23.

Proof of Theorem 23. For convenience set $\bar{\mathcal{L}}_t(x, y) = x^\top A_t y$. Let (x^*, y^*) be any saddle point of $\min_{x\in\Delta}\max_{y\in\Delta}\sum_{t=1}^{T} x^\top A_t y$, let (x_p^*, y_p^*) be the respective projections onto Δ_θ using $\|\cdot\|_\infty$ norm. By the choice of θ we have that $|\ln(\theta)| > 1$ additionally, notice that $\max_{z\in\Delta_\theta}\sum_{i=1}^{d} z_i \ln(z_i) + \ln(d) \leq 0 + \ln(d)$ by Jensen's inequality.

$$
\begin{aligned}
&\sum_{t=1}^{T} x_t^\top A_t y_t - \min_{x\in\Delta}\max_{y\in\Delta}\sum_{t=1}^{T} x^\top A_t y \\
&\leq \sum_{t=1}^{T} x_t^\top A_t y_t - \min_{x\in\Delta_\theta}\max_{y\in\Delta_\theta}\sum_{t=1}^{T} x^\top A_t y + G_{\bar{\mathcal{L}}} T\|x^* - x_p^*\|_1 \quad \text{by Lemma 39} \\
&\leq \sum_{t=1}^{T} x_t^\top A_t y_t - \min_{x\in\Delta_\theta}\max_{y\in\Delta_\theta}\sum_{t=1}^{T} x^\top A_t y + 2G_{\bar{\mathcal{L}}} T\theta(d_1 - 1) \quad \text{by Lemma 38} \\
&\leq 8\eta[G_{\bar{\mathcal{L}}} + \frac{1}{\eta}\max(G_{R_X}, G_{R_Y})]^2(1 + \ln(T)) + \frac{T}{\eta}\max_{y\in\Delta_\theta} R_Y(y) + \frac{T}{\eta}\max_{x\in\Delta_\theta} R_X(x) \\
&\qquad + 2G_{\bar{\mathcal{L}}} T\theta(d_1 - 1) \quad \text{by Theorem 22} \\
&\leq 8\eta[G_{\bar{\mathcal{L}}} + \frac{|\ln(\theta)|}{\eta}]^2(1 + \ln(T)) + \frac{T}{\eta}\max_{y\in\Delta_\theta} R_Y(y) + \frac{T}{\eta}\max_{x\in\Delta_\theta} R_X(x) + 2G_{\bar{\mathcal{L}}} T\theta(d_1 - 1) \\
&\leq 32\eta G_{\bar{\mathcal{L}}}^2(1 + \ln(T)) + \frac{T}{\eta}\max_{y\in\Delta_\theta} R_Y(y) + \frac{T}{\eta}\max_{x\in\Delta_\theta} R_X(x) + 2G_{\bar{\mathcal{L}}} T e^{-\eta G_{\bar{\mathcal{L}}}}(d_1 - 1) \\
&\qquad \text{by the choice of } \theta \\
&\leq 32\eta G_{\bar{\mathcal{L}}}^2(1 + \ln(T)) + \frac{T}{\eta}\ln(d_2) + \frac{T}{\eta}\ln(d_1) + 2G_{\bar{\mathcal{L}}} T e^{-\eta G_{\bar{\mathcal{L}}}}(d_1 - 1) \\
&\leq 32 G_{\bar{\mathcal{L}}}\sqrt{T}(1 + \ln(T)) + \sqrt{T}(\ln d_1 + \ln d_2) + 2d_1 G_{\bar{\mathcal{L}}} T e^{-\sqrt{T}} \\
&= O\left(\ln(T)\sqrt{T} + \sqrt{T}\max\{\ln d_1, \ln d_2\}\right) + \quad o(1)\max\{d_1, d_2\}.
\end{aligned}
$$

The last line follows because $G_{\bar{\mathcal{L}}} \leq 1$, since each entry of A is bounded between $[-1, 1]$. A symmetrical argument yields the other side of the inequality. □

4.5 Online Matrix Games: Bandit Feedback

In this section we focus on the OMG problem under bandit feedback. In this setting, the players observe in every round only the payoff corresponding to the chosen actions. If Player 1 chooses action i, Player 2 chooses action j, and the payoff matrix at that time step is A_t, then the players observe only $(A_t)_{ij}$ instead of the full matrix A_t. The limited feedback makes the problem significantly more challenging than the full information one: the players must find a way to *exploit* (use all previous information to try to play a Nash Equilibrium) and *explore* (try to estimate A_t in every round). This problem resembles that of Online Bandit Optimization [14, 33, 53, 70], while the main difference is that with one function evaluation we must estimate a matrix A_t instead of the gradients $\nabla_x \mathcal{L}_t(x, y)$ and $\nabla_y \mathcal{L}_t(x, y)$ where $\mathcal{L}_t = x^\top A_t y$.

Before proceeding we establish some useful notation. For $i = 1, ..., d$, let $e_i \in \mathbb{R}^d$ be the collection of standard unit vectors i.e. e_i is the vector that has a 1 in the i-th entry and 0 in the rest. Let $e_{x,t}$ be the standard unit vector corresponding to the decision made by Player 1 for round t, define $e_{y,t}$ similarly. Notice that under bandit feedback, in round t both players only observe the quantity $e_{x,t}^\top A_t e_{y,t}$.

4.5.1 A One-Point Estimate for $\mathcal{L}(x, y) = x^\top A y$

As explained previously, in each round t the players must estimate A_t by observing only one of its entries. To this end, we allow the players to share with each other their decisions and to randomize *jointly* (a similar assumption is used to define correlated equilibria in zero-sum games, see [16]). The following result shows how to build a random estimate of A by observing only one of its entries.

Theorem 24. *Let $x \in \Delta_{X,\delta}, y \in \Delta_{Y,\delta}$ with $d_1, d_2 \geq 2$ and $\delta > 0$. Sample $i' \sim x, j' \sim y$. Let $\hat{A}$ be the $d_1 \times d_2$ matrix with $\hat{A}_{i,j} = 0$ for all i, j such that $i \neq i'$ and $j \neq j'$ and*

$\hat{A}_{i',j'} = \frac{A_{i',j'}}{x(i')y(j')}$. *It holds that*

$$\mathbb{E}_{i'\sim x, j'\sim y}[\hat{A}] = A.$$

4.5.2 Bandit Online Matrix Games RFTL

We now present an algorithm that ensures sublinear (i.e. $o(T)$) NE regret under bandit feedback for the OMG problem that holds against an adaptive adversary. By adaptive adversary, we mean that the payoff matrices A_t can depend on the players' actions up to time $t-1$; in particular, we assume the adversary does not observe the actions chosen by the players for time period t when choosing A_t. We consider an algorithm that runs OMG-RFTL on a sequence of functions $\hat{\mathcal{L}}_t \triangleq x^\top \hat{A}_t y$, where $\hat{A}_t$ is the unbiased one-point estimate of A_t derived in Theorem 24. Recall that the iterates of OMG-RFTL algorithm are distributions over the possible actions of both players. In order to generate the estimate $\hat{A}_t$, both players will sample an action from their distributions and weigh their observation with the inverse probability of obtaining that observation.

Algorithm 11 Bandit Online-Matrix-Games Regularized-Follow-the-Leader (BANDIT-OMG-RFTL)

input: $x_1 \in \Delta_{X,\delta} \subset \mathbb{R}^{d_1}$, $y_1 \in \Delta_{Y,\delta} \subset \mathbb{R}^{d_2}$, parameters: $\eta > 0$, $0 < \delta < \min\{\frac{1}{d_1}, \frac{1}{d_2}\}$.
for $t = 1, ...T$ **do**
  Sample independently $e_{x,t} \sim x_t$ and $e_{y,t} \sim y_t$
  Observe $e_{x,t}^\top A_t e_{y,t}$
  Build $\hat{A}_t$ as in Theorem 24 using $e_{x,t}^\top A_t e_{y,t}, x_t, y_t$
  $\hat{\mathcal{L}}_t \leftarrow x^\top \hat{A}_t y$
  $\mathcal{L}_t(x,y) \leftarrow \hat{\mathcal{L}}_t + \frac{1}{\eta} R_X(x) - \frac{1}{\eta} R_Y(y)$
  $x_{t+1} \leftarrow \arg\min_{x\in\Delta_{X,\theta}} \max_{y\in\Delta_{Y,\theta}} \sum_{\tau=1}^{t} \mathcal{L}_t(x,y)$
  $y_{t+1} \leftarrow \arg\max_{y\in\Delta_{Y,\theta}} \min_{x\in\Delta_{X,\theta}} \sum_{\tau=1}^{t} \mathcal{L}_t(x,y)$
end for

We have the following guarantee for BANDIT-OMG-RFTL.

Theorem 25. *Let $\{A_t\}_{t=1}^T$ be any sequence of payoff matrices chosen by an adaptive adversary. Let $\{e_{x,t}, e_{y,t}\}_{t=1}^T$ be the iterates generated by* BANDIT-OMG-FTRL. *Setting*

$\delta = \frac{1}{T^{1/6}}$, $\eta = T^{1/6}$ *ensures*

$$\begin{aligned}&\left|\mathbb{E}\left[\sum_{t=1}^{T} e_{x,t}^{\top} A_t e_{y,t} - \min_{x\in X}\max_{y\in Y}\sum_{t=1}^{T} x^{\top} A_t y\right]\right| \\ \leq & O((d_1+d_2)\ln(T)T^{5/6})\end{aligned}$$

where the expectation is taken with respect to randomization in the algorithm.

The full proof of this Theorem will be given shortly. We now give a sketch of the proof. The total payoff given to each of the players is given by $\sum_{t=1}^{T} e_{x,t}^{\top} A_t e_{y,t}$ so we must relate this quantity to the iterates $\{x_t, y_t\}_{t=1}^{T}$ of OMG-RFTL when run on sequence of matrices $\{\hat{A}_t\}_{t=1}^{T}$. The following two lemmas will allow us to do so.

Lemma 40. *Let $\{e_{x,t}, e_{y,t}\}_{t=1}^{T}$ be the sequence of iterates generated by* BANDIT-OMG-RFTL. *It holds that*

$$\mathbb{E}\left[\sum_{t=1}^{T} e_{x,t}^{\top} A_t e_{y,t}\right] = \mathbb{E}\left[\sum_{t=1}^{T} x_t^{\top} A_t y_t\right],$$

where the expectation is taken with respect to the internal randomness of the algorithm.

Proof of Lemma 40.

$$\begin{aligned}&\mathbb{E}[\sum_{t=1}^{T} e_{x,t}^{\top} A_t e_{y,t}] \\ &= \mathbb{E}[\sum_{t=1}^{T-1} e_{x,t}^{\top} A_t e_{y,t}] + \mathbb{E}[e_{x,T}^{\top} A_T e_{y,T}] \\ &= \mathbb{E}[\sum_{t=1}^{T-1} e_{x,t}^{\top} A_t e_{y,t}] + \mathbb{E}[\mathbb{E}_{e_{x,T}\sim x_t, e_{y,T}\sim y_t}[e_{x,T}^{\top} A_T e_{y,T} | \tau = 1, ..., T-1]] \\ &= \mathbb{E}[\sum_{t=1}^{T-1} e_{x,t}^{\top} A_t e_{y,t}] + \mathbb{E}[x_T^{\top}\mathbb{E}_{e_{x,T}\sim x_t, e_{y,T}\sim y_t}[A_T | \tau = 1, ..., T-1] y_T] \\ &= \mathbb{E}[\sum_{t=1}^{T-1} x_t^{\top} A_t y_t] + \mathbb{E}[x_T^{\top} A_T y_T].\end{aligned}$$

Repeating the argument $T-1$ more times yields the result. □

Lemma 41. *It holds that*

$$\mathbb{E}\left[\sum_{t=1}^{T} x_t^\top \hat{A}_t y_t\right] = \mathbb{E}\left[\sum_{t=1}^{T} x_t^\top A_t y_t\right],$$

where the expectation is with respect to all the internal randomness of the algorithm.

Proof of Lemma 41.

$$\begin{aligned}
&\mathbb{E}[\sum_{t=1}^{T} x_t^\top \hat{A}_t y_t] \\
&= \mathbb{E}[\sum_{t=1}^{T-1} x_t^\top \hat{A}_t y_t] + \mathbb{E}[x_T^\top \hat{A}_T y_T] \\
&= \mathbb{E}[\sum_{t=1}^{T-1} x_t^\top \hat{A}_t y_t] + \mathbb{E}[\mathbb{E}[x_T^\top \hat{A}_T y_T | \tau = 1, ..., T-1]] \\
&= \mathbb{E}[\sum_{t=1}^{T-1} x_t^\top \hat{A}_t y_t] + \mathbb{E}[x_T^\top \mathbb{E}[\hat{A}_T | \tau = 1, ..., T-1] y_T] \\
&= \mathbb{E}[\sum_{t=1}^{T-1} x_t^\top \hat{A}_t y_t] + \mathbb{E}[x_T^\top A_T y_T] \quad \text{by Theorem 24.}
\end{aligned}$$

Repeating the argument $T-1$ more times yields the result. □

We will then bound the difference between the comparator term $\min_{x\in\Delta}\max_{y\in\Delta}\sum_{t=1}^{T} x^\top A_t y$ and the comparator term Theorem 23 gives us by running OMG-RFTL on functions $\{\hat{\mathcal{L}}\}_{t=1}^{T}$, $\min_{x\in\Delta}\max_{y\in\Delta}\sum_{t=1}^{T} x^\top \hat{A}_t y$. Special care must be taken to ensure this difference holds even against an adaptive adversary. To this end, we use the next two lemmas; in particular, the proof of Lemma 43 relies heavily on Theorem 24.

Lemma 42. *With probability 1 it holds that*

$$\left|\min_{x\in\Delta_X^\delta}\max_{y\in\Delta_Y^\delta}\sum_{t=1}^{T} x^\top A_t y - \min_{x\in\Delta_X^\delta}\max_{y\in\Delta_Y^\delta}\sum_{t=1}^{T} x^\top \hat{A}_t y\right|$$

$$\leq \left\| \sum_{t=1}^{T} A_t y - \hat{A}_t y \right\|_2 .$$

Proof of Lemma 42. Let us fist bound $|\sum_{t=1}^{T} x^\top A_t y - \sum_{t=1}^{T} x^\top \hat{A}_t y|$ for any $x \in \Delta_X$ and $y \in \Delta_Y$ with probability 1.

$$\begin{aligned}
&|\sum_{t=1}^{T} x^\top A_t y - \sum_{t=1}^{T} x^\top \hat{A}_t y| \\
&= |x^\top (\sum_{t=1}^{T} A_t y - \sum_{t=1}^{T} \hat{A}_t y)| \\
&\leq \|x\|_2 \| \sum_{t=1}^{T} A_t y - \hat{A}_t y\|_2 \\
&\leq \| \sum_{t=1}^{T} A_t y - \hat{A}_t y\|_2
\end{aligned}$$

It now follows that

$$\begin{aligned}
&\sum_{t=1}^{T} x^\top \hat{A}_t y \leq \sum_{t=1}^{T} x^\top A_t y + \| \sum_{t=1}^{T} A_t y - \hat{A}_t y\|_2 \\
\implies &\min_{x \in \Delta_{X,\delta}} \sum_{t=1}^{T} x^\top \hat{A}_t y \leq \sum_{t=1}^{T} x^\top A_t y + \| \sum_{t=1}^{T} A_t y - \hat{A}_t y\|_2 \quad \forall x \in \Delta_{X,\delta}, y \in \Delta_{Y,\delta} \\
\implies &\min_{x \in \Delta_{X,\delta}} \sum_{t=1}^{T} x^\top \hat{A}_t y \leq \max_{y \in \Delta_{Y,\delta}} \sum_{t=1}^{T} x^\top A_t y + \| \sum_{t=1}^{T} A_t y - \hat{A}_t y\|_2 \quad \forall x \in \Delta_{X,\delta}, y \in \Delta_{Y,\delta} \\
\implies &\max_{y \in \Delta_{Y,\delta}} \min_{x \in \Delta_{X,\delta}} \sum_{t=1}^{T} x^\top \hat{A}_t y \leq \min_{x \in \Delta_{X,\delta}} \max_{y \in \Delta_{Y,\delta}} \sum_{t=1}^{T} x^\top A_t y + \| \sum_{t=1}^{T} A_t y - \hat{A}_t y\|_2, \\
&\qquad \forall x \in \Delta_{X,\delta}, y \in \Delta_{Y,\delta}.
\end{aligned}$$

This concludes the proof as $\max_{y \in \Delta_{Y,\delta}} \min_{x \in \Delta_{X,\delta}} \sum_{t=1}^{T} x^\top \hat{A}_t y = \min_{x \in \Delta_{X,\delta}} \max_{y \in \Delta_{Y,\delta}} \sum_{t=1}^{T} x^\top \hat{A}_t y$ (since the function is convex-concave and the sets Δ_Y^δ and Δ_X^δ are convex and compact), the other side of the inequality can be obtained using the other inequality follows from applying the same reasoning. □

Lemma 43. *It holds that*

$$\mathbb{E}\left[\left\|\sum_{t=1}^{T} A_t y - \hat{A}_t y\right\|_2\right] \leq \frac{2\sqrt{T}\min(d_1, d_2)}{\delta^2},$$

where the expectation is taken with respect to the internal randomness of the algorithm.

Proof of Lemma 43. For any y define $\alpha_t \triangleq A_t y - \hat{A}_t y$. We first show that for all t, t' such that $t < t'$ it holds that $\mathbb{E}[\alpha_t^\top \alpha_{t'}] = 0$. Indeed

$$\begin{aligned}
\mathbb{E}[\alpha_t^\top \alpha_{t'}] &= \mathbb{E}[(A_t y - \hat{A}_t y)^\top (A_{t'} y - \hat{A}_{t'} y)] \\
&= \mathbb{E}[(A_t y)^\top A_{t'} y - (A_t y)^\top \hat{A}_{t'} y - (\hat{A}_t y)^\top A_{t'} y + (\hat{A}_t y)^\top \hat{A}_{t'} y] \\
&= (A_t y)^\top A_{t'} y - (A_t y)^\top A_{t'} y - (A_t y)^\top A_{t'} y + \mathbb{E}[(\hat{A}_t y)^\top \hat{A}_{t'} y] \\
&= (A_t y)^\top A_{t'} y - (A_t y)^\top A_{t'} y - (A_t y)^\top A_{t'} y + (A_t y)^\top A_{t'} y \\
&= 0,
\end{aligned}$$

where the second to last line follows since

$$\begin{aligned}
\mathbb{E}[(\hat{A}_t y)^\top \hat{A}_{t'} y] &= \mathbb{E}_{1,\ldots,t'-1}[\mathbb{E}[(\hat{A}_t y)^\top \hat{A}_{t'} y | \tau = 1, \ldots, t'-1]] \\
&= \mathbb{E}_{1,\ldots,t'-1}[(\hat{A}_t y)^\top \mathbb{E}[\hat{A}_{t'} y | \tau = 1, \ldots, t'-1]] \\
&= \mathbb{E}_{1,\ldots,t'-1}[(\hat{A}_t y)^\top A_{t'} y] \\
&= (A_t y)^\top A_{t'} y.
\end{aligned}$$

Now,

$$\begin{aligned}
\mathbb{E}[\|\sum_{t=1}^{T} A_t y - \hat{A}_t y\|_2] &= \sqrt{\mathbb{E}[\|\sum_{t=1}^{T} \alpha_t\|_2]^2} \\
&\leq \sqrt{\mathbb{E}[\|\sum_{t=1}^{T} \alpha_t\|_2^2]} \quad \text{by Jensen's Inequality}
\end{aligned}$$

$$= \sqrt{\sum_{t=1}^{T} \mathbb{E}[\|\alpha_t\|_2^2] + 2\sum_{t<t'} \mathbb{E}[\alpha_t^\top \alpha_{t'}]}$$
$$= \sqrt{\sum_{t=1}^{T} \mathbb{E}[\|A_t y - \hat{A}_t y\|_2^2]}$$
$$\leq \sqrt{\sum_{t=1}^{T} \mathbb{E}[2\|A_t y\|^2 + 2\|\hat{A}_t y\|_2^2]}$$

We proceed to bound $\|\hat{A}_t y\|_2$, the upper bound we obtain will also bound $\|A_t y\|$ because of the following fact. If the random vector $\tilde{a}$ satisfies $\|\tilde{a}\| \leq c$ for some constant c with probability 1 then $\|\mathbb{E}\tilde{a}\| \leq c$. Indeed by Jensen's inequality we have that $\|\mathbb{E}\tilde{a}\| \leq \mathbb{E}\|\tilde{a}\| \leq c$. Let us omit the subscript t for the rest of the proof. Let $\hat{A}_{[i,:]}$ be the i-th row of matrix $\hat{A}$.

$$\begin{aligned}
\|\hat{A}y\|_2 &= \sqrt{\sum_{i=1}^{d_1} \big[\sum_{j=1}^{d_2} \hat{a}_{i,j} y_j\big]^2} \\
&\leq \sum_{i=1}^{d_1} \sqrt{\big[\sum_{j=1}^{d_2} \hat{a}_{i,j} y_j\big]^2} \\
&= \sum_{i=1}^{d_1} \big|\sum_{j=1}^{d_2} \hat{a}_{i,j} y_j\big| \\
&\leq \sum_{i=1}^{d_1} \|\hat{A}_{[i,:]}\|_\infty \|y\|_1 \quad \text{by generalized Cauchy Schwartz} \\
&\leq d_1 \max_{i,j} |\frac{A_{i,j}}{\delta^2}| \quad \text{by definition of } \hat{A} \text{ and using the fact that } x_t \in \Delta_{X,\delta} \text{ and } y_t \in \Delta_{Y,\delta} \\
&\leq \frac{d_1}{\delta^2}.
\end{aligned}$$

Notice the upper bound $\frac{d_2}{\delta^2}$ can also be obtained by interchanging the summations and repeating the argument. This yields the desired result. □

The proof of Theorem 25 follows by combining Lemmas 40 through 43, with careful choice of tuning parameters.

Proof of Theorem 25. We first focus on one side of the inequality,

$$
\begin{aligned}
&\mathbb{E}[\sum_{t=1}^{T} e_{x,t}^{\top} A_t e_{y,t} - \min_{x\in X}\max_{y\in Y}\sum_{t=1}^{T} x^{\top} A_t y] \\
&= \mathbb{E}[\sum_{t=1}^{T} e_{x,t}^{\top} A_t e_{y,t}] - \mathbb{E}[\min_{x\in X}\max_{y\in Y}\sum_{t=1}^{T} x^{\top} A_t y] \\
&= \mathbb{E}[\sum_{t=1}^{T} x_t^{\top} A_t y_t] - \mathbb{E}[\min_{x\in X}\max_{y\in Y}\sum_{t=1}^{T} x^{\top} A_t y] \quad \text{by Lemma 40} \\
&= \mathbb{E}[\sum_{t=1}^{T} x_t^{\top} A_t y_t] - \mathbb{E}[\min_{x\in \Delta_X^{\delta}}\max_{y\in \Delta_Y^{\delta}}\sum_{t=1}^{T} x^{\top} A_t y] + 2\delta G_{\bar{\mathcal{L}}}^{\|\cdot\|_1}(d_1-1)T \\
&\quad \text{by Lemmas 38 and 39} \\
&\leq \mathbb{E}[\sum_{t=1}^{T} x_t^{\top} A_t y_t] - \mathbb{E}[\min_{x\in \Delta_X^{\delta}}\max_{y\in \Delta_Y^{\delta}}\sum_{t=1}^{T} x^{\top} \hat{A}_t y] + \frac{2\sqrt{T}\min(d_1,d_2)}{\delta^2} + 2\delta G_{\bar{\mathcal{L}}}^{\|\cdot\|_1}(d_1-1)T \\
&\quad \text{by Lemmas 42 and 43} \\
&\leq \mathbb{E}[\sum_{t=1}^{T} x_t^{\top} \hat{A}_t y_t] - \mathbb{E}[\min_{x\in \Delta_X^{\delta}}\max_{y\in \Delta_Y^{\delta}}\sum_{t=1}^{T} x^{\top} \hat{A}_t y] + \frac{2\sqrt{T}\min(d_1,d_2)}{\delta^2} + 2\delta G_{\bar{\mathcal{L}}}^{\|\cdot\|_1}(d_1-1)T \\
&\quad \text{by Lemma 41} \\
&\leq 8\eta[G_{\hat{\mathcal{L}}}^{\|\cdot\|_1} + \frac{|\ln(\delta)|}{\eta}]^2(1+\ln(T)) + \frac{T}{\eta}(\ln(d_1)+\ln(d_2)) \\
&\quad + \frac{2\sqrt{T}\min(d_1,d_2)}{\delta^2} + 2\delta G_{\bar{\mathcal{L}}}^{\|\cdot\|_1}(d_1-1)T \quad \text{as in the proof of Theorem 23} \\
&= 8\eta[\frac{1}{\delta^2} + \frac{|\ln(\delta)|}{\eta}]^2(1+\ln(T)) + \frac{T}{\eta}(\ln(d_1)+\ln(d_2)) + \frac{2\sqrt{T}\min(d_1,d_2)}{\delta^2} \\
&\quad + 2\delta(d_1-1)T \quad \text{by Lemma 33} \\
&= O((d_1+d_2)\ln(T)T^{5/6}) \quad \text{after plugging in } \delta = \frac{1}{T^{1/6}},\ \eta = T^{1/6}
\end{aligned}
$$

The other side of the inequality follows by a symmetrical argument. □

4.6 The Strongly Convex-Concave Case

We now present an algorithm for the case where the payoff functions $\{\mathcal{L}_t\}_{t=1}^{T}$ are strongly convex-concave. We show that the following simple algorithm Saddle-Point Follow-the-

Leader (SP-FTL), which is a variant of the Follow-the-Leader (FTL) algorithm by Kalai and Vempala [82], attains sublinear NE regret.

Algorithm 12 Saddle-Point Follow-the-Leader (SP-FTL)

input: $x_1 \in X$, $y_1 \in Y$
for $t = 1, ..., T$ **do**
  Choose actions (x_t, y_t)
  Observe function $\mathcal{L}_t$
  Set $x_{t+1} \leftarrow \arg\min_{x \in X} \max_{y \in Y} \sum_{\tau=1}^{t} \mathcal{L}_\tau(x, y)$
  Set $y_{t+1} \leftarrow \arg\max_{y \in Y} \min_{x \in X} \sum_{\tau=1}^{t} \mathcal{L}_\tau(x, y)$
end for

The main difference between SP-FTL and FTL is that in SP-FTL both players update jointly and play the (unique) saddle point of the sum of the games observed so far. In contrast, the updates for Follow-the-Leader would be $x_{t+1}^{FTL} \leftarrow \arg\min_{x \in X} \sum_{\tau=1}^{t} \mathcal{L}_\tau(x, y_\tau^{FTL})$ and $y_{t+1}^{FTL} \leftarrow \arg\max_{y \in Y} \sum_{\tau=1}^{t} \mathcal{L}_\tau(x_\tau^{FTL}, y)$ for $t = 2, ..., T$ and x_1^{FTL}, y_1^{FTL} are arbitrarily chosen from their respective sets X and Y. It is easy to see that the sequence of iterates is in general not the same.

Theorem 26. *Let $\{\mathcal{L}_t(x, y)\}_{t=1}^{T}$ be an arbitrary sequence of H-strongly convex-concave, G-Lipschitz functions. Then, the* SP-FTL *algorithm guarantees*

$$\mathsf{SP\text{-}Regret}(T) = \left| \sum_{t=1}^{T} \mathcal{L}_t(x_t, y_t) - \min_{x \in X} \max_{y \in Y} \sum_{t=1}^{T} \mathcal{L}_t(x, y) \right| \leq \frac{8G^2}{H}(1 + \log T).$$

The proof of Theorem 26 is based on the following two lemmas. We first analyze a quantity that is similar to SP-regret, but with actions (x_t, y_t) replaced by (x_{t+1}, y_{t+1}) (Lemma 44). This analysis framework is known as the Follow-the-Leader vs. Be-the-Leader scheme [82]. We then show that consecutive iterates of SP-FTL have distances diminishing in the order of $O(1/t)$. The proof heavily utilizes the KKT conditions associated with points (x_t, y_t) and (x_{t+1}, y_{t+1}) (Lemma 45).

Lemma 44. *Let* $\{(x_t, y_t)\}_{t=1}^T$ *be the iterates of* SP-FTL. *It holds that*

$$-G\sum_{t=1}^{T}\|x_t - x_{t+1}\| \le \sum_{t=1}^{T}\mathcal{L}_t(x_{t+1}, y_{t+1}) - \min_{x\in X}\max_{y\in Y}\sum_{t=1}^{T}\mathcal{L}_t(x,y) \le G\sum_{t=1}^{T}\|y_t - y_{t+1}\|. \tag{4.11}$$

Proof. We first prove the second inequality. We proceed by induction. The base case $t = 1$ holds by definition of (x_2, y_2), indeed

$$\mathcal{L}_1(x_2, y_2) + G\|y_1 - y_2\| \ge \mathcal{L}_1(x_2, y_2) := \min_{x\in X}\max_{y\in Y}\mathcal{L}_1(x,y).$$

We now assume the following claim holds for $T - 1$:

$$\min_{x\in X}\max_{y\in Y}\sum_{t=1}^{T-1}\mathcal{L}_t(x,y) \ge \sum_{t=1}^{T-1}\mathcal{L}_t(x_{t+1}, y_{t+1}) - G\sum_{t=1}^{T-1}\|y_t - y_{t+1}\|, \tag{4.12}$$

and show it holds for T.

$$\begin{aligned}
&\min_{x\in X}\max_{y\in Y}\sum_{t=1}^{T}\mathcal{L}_t(x,y)\\
&= \sum_{t=1}^{T-1}\mathcal{L}_t(x_{T+1}, y_{T+1}) + \mathcal{L}_T(x_{T+1}, y_{T+1})\\
&\ge \sum_{t=1}^{T-1}\mathcal{L}_t(x_{T+1}, y_T) + \mathcal{L}_T(x_{T+1}, y_T) && \text{by Equation (4.2)}\\
&\ge \sum_{t=1}^{T-1}\mathcal{L}_t(x_T, y_T) + \mathcal{L}_T(x_{T+1}, y_T) && \text{by Equation (4.2)}\\
&\ge \sum_{t=1}^{T-1}\mathcal{L}_t(x_{t+1}, y_{t+1}) - G\sum_{t=1}^{T-1}\|y_t - y_{t+1}\| + \mathcal{L}_T(x_{T+1}, y_T) && \text{by Equation (4.12)}\\
&= \sum_{t=1}^{T}\mathcal{L}_t(x_{t+1}, y_{t+1}) - G\sum_{t=1}^{T-1}\|y_t - y_{t+1}\| + \mathcal{L}_T(x_{T+1}, y_T) - \mathcal{L}_T(x_{T+1}, y_{T+1})\\
&\ge \sum_{t=1}^{T}\mathcal{L}_t(x_{t+1}, y_{t+1}) - G\sum_{t=1}^{T-1}\|y_t - y_{t+1}\| - G\|y_T - y_{T+1}\|\\
&= \sum_{t=1}^{T}\mathcal{L}_t(x_{t+1}, y_{t+1}) - G\sum_{t=1}^{T}\|y_t - y_{t+1}\|.
\end{aligned}$$

We now show by induction that

$$\min_{x\in X}\max_{y\in Y}\sum_{t=1}^{T}\mathcal{L}_t(x,y)\le\sum_{t=1}^{T}\mathcal{L}_t(x_{t+1},y_{t+1})+G\sum_{t=1}^{T}\|x_t-x_{t+1}\|.$$

Indeed, $t=1$ follows from the definition of (x_2,y_2). We now assume the claim holds for $T-1$ and prove it for T:

$$\begin{aligned}
&\min_{x\in X}\max_{y\in Y}\sum_{t=1}^{T}\mathcal{L}_t(x,y)\\
&=\sum_{t=1}^{T}\mathcal{L}_t(x_{T+1},y_{T+1})\\
&\le\sum_{t=1}^{T-1}\mathcal{L}_t(x_T,y_{T+1})+\mathcal{L}_T(x_T,y_{T+1}) && \text{by Equation (4.2)}\\
&\le\sum_{t=1}^{T-1}\mathcal{L}_t(x_T,y_T)+\mathcal{L}_T(x_T,y_{T+1}) && \text{by Equation (4.2)}\\
&\le\sum_{t=1}^{T-1}\mathcal{L}_t(x_{t+1},y_{t+1})+G\sum_{t=1}^{T-1}\|x_t-x_{t+1}\|\\
&\qquad+\mathcal{L}_T(x_T,y_{T+1})+\mathcal{L}_T(x_{T+1},y_{T+1})-\mathcal{L}_T(x_{T+1},y_{T+1}) && \text{by induction claim}\\
&\le\sum_{t=1}^{T}\mathcal{L}_t(x_{t+1},y_{t+1})+G\sum_{t=1}^{T}\|x_t-x_{t+1}\| && \text{since } \mathcal{L}_T \text{ is } G\text{-Lipschitz.}
\end{aligned}$$

□

Lemma 45. *Let* $\{(x_t,y_t)\}_{t=1}^{T}$ *be the iterates of* SP-FTL. *It holds that*

$$||x_t-x_{t+1}||+||y_t-y_{t+1}||\le\frac{4G}{Ht}.$$

Proof. Fix t. Define

$$J(x,y)\triangleq\sum_{\tau=1}^{t-1}\mathcal{L}_\tau(x,y)+\mathcal{L}_t(x,y)$$

so that $(x_{t+1}, y_{t+1}) = \min_{x \in X} \max_{y \in Y} J(x, y)$. Since J is Ht-strongly convex it holds that for any $x \in X$ and any $y \in Y$

$$J(x, y) \geq J(x_{t+1}, y) + \nabla_x J(x_{t+1}, y)^\top (x - x_{t+1}) + \frac{Ht}{2} \|x - x_{t+1}\|^2.$$

Plugging in $y = y_{t+1}$ and recalling the KKT condition $\nabla_x J(x_{t+1}, y_{t+1})^\top (x - x_{t+1}) \geq 0$, we have that for any $x \in X$

$$\frac{2}{Ht} \big[J(x, y_{t+1}) - J(x_{t+1}, y_{t+1}) \big] \geq \|x - x_{t+1}\|^2. \tag{4.13}$$

Similarly, since J is Ht strongly concave. That is, for any $y \in Y$

$$J(x_{t+1}, y) \leq J(x_{t+1}, y_{t+1}) + \nabla_y J(x_{t+1}, y_{t+1})^\top (y - y_{t+1}) - \frac{Ht}{2} \|y - y_{t+1}\|^2.$$

Together with the KKT condition $\nabla_y J(x_{t+1}, y_{t+1})^\top (y - y_{t+1}) \leq 0$ we get that for any $y \in Y$

$$\frac{2}{Ht} \big[J(x_{t+1}, y_{t+1}) - J(x_{t+1}, y) \big] \geq \|y - y_{t+1}\|^2. \tag{4.14}$$

Adding up Equations (4.13) and (4.14), plugging $x = x_t$ and $y = y_t$ we get

$$\begin{aligned}
&\frac{2}{Ht} \big[J(x_t, y_{t+1}) - J(x_{t+1}, y_t) \big] \geq \|x_t - x_{t+1}\|^2 + \|y_t - y_{t+1}\|^2 \\
\iff &\frac{2}{Ht} \big[\sum_{\tau=1}^{t-1} \mathcal{L}_\tau(x_t, y_{t+1}) + \mathcal{L}_t(x_t, y_{t+1}) - [\sum_{\tau=1}^{t-1} \mathcal{L}_\tau(x_{t+1}, y_t) + \mathcal{L}_t(x_{t+1}, y_t)] \big] \\
&\qquad \geq \|x_t - x_{t+1}\|^2 + \|y_t - y_{t+1}\|^2 \\
\implies &\frac{2}{Ht} \big[\sum_{\tau=1}^{t-1} \mathcal{L}_\tau(x_t, y_t) + \mathcal{L}_t(x_t, y_{t+1}) - [\sum_{\tau=1}^{t-1} \mathcal{L}_\tau(x_{t+1}, y_t) + \mathcal{L}_t(x_{t+1}, y_t)] \big] \\
&\qquad \geq \|x_t - x_{t+1}\|^2 + \|y_t - y_{t+1}\|^2.
\end{aligned}$$

since $\sum_{\tau=1}^{t-1} \mathcal{L}_\tau(x_t, y_{t+1}) \leq \sum_{\tau=1}^{t-1} \mathcal{L}_\tau(x_t, y_t)$.

Additionally, since $\sum_{\tau=1}^{t-1} \mathcal{L}_\tau(x_t, y_t) \leq \sum_{\tau=1}^{t-1} \mathcal{L}_\tau(x_{t+1}, y_t)$ we have

$$\begin{aligned}
&\frac{2}{Ht}\big[\sum_{\tau=1}^{t-1} \mathcal{L}_\tau(x_t, y_t) + \mathcal{L}_t(x_t, y_{t+1}) - \sum_{\tau=1}^{t-1} \mathcal{L}_\tau(x_t, y_t) - \mathcal{L}_t(x_{t+1}, y_t)\big] \\
&\qquad \geq \|x_t - x_{t+1}\|^2 + \|y_t - y_{t+1}\|^2 \\
\iff &\frac{2}{Ht}\big[\mathcal{L}_t(x_t, y_{t+1}) - \mathcal{L}_t(x_{t+1}, y_t)\big] \geq \|x_t - x_{t+1}\|^2 + \|y_t - y_{t+1}\|^2 \\
\implies &\frac{2}{Ht} G\|[x_t; y_{t+1}] - [x_{t+1}; y_t]\| \geq \|x_t - x_{t+1}\|^2 + \|y_t - y_{t+1}\|^2 \\
\implies &\frac{2}{Ht} G\big[\|x_t - x_{t+1}\| + \|y_t - y_{t+1}\|\big] \geq \|x_t - x_{t+1}\|^2 + \|y_t - y_{t+1}\|^2 \\
\implies &\frac{2G}{Ht} \geq \frac{\|x_t - x_{t+1}\|^2 + \|y_t - y_{t+1}\|^2}{\|x_t - x_{t+1}\| + \|y_t - y_{t+1}\|}.
\end{aligned}$$

Now, since x^2 is a convex function $\frac{a^2}{2} + \frac{b^2}{2} \geq \left(\frac{a+b}{2}\right)^2$ therefore $a^2 + b^2 \geq \frac{(a+b)^2}{2}$. This, together with the last implication, yields the result

$$\frac{4G}{Ht} \geq \|x_t - x_{t+1}\| + \|y_t - y_{t+1}\|.$$

□

Now we are ready to prove Theorem 26.

Proof of Theorem 26. We have

$$\begin{aligned}
&\sum_{t=1}^{T} \mathcal{L}_t(x_t, y_t) - \min_{x \in X} \max_{y \in Y} \sum_{t=1}^{T} \mathcal{L}_t(x, y) \\
&\leq \sum_{t=1}^{T} \mathcal{L}_t(x_t, y_t) - \sum_{t=1}^{T} \mathcal{L}_t(x_{t+1}, y_{t+1}) + G \sum_{t=1}^{T} \|y_t - y_{t+1}\| && \text{by Lemma 44} \\
&\leq \sum_{t=1}^{T} G\|[x_t; y_t] - [x_{t+1}; y_{t+1}]\| + G \sum_{t=1}^{T} \|y_t - y_{t+1}\| && \text{since } \mathcal{L}_t \text{ is } G\text{-Lipschitz} \\
&\leq G \sum_{t=1}^{T} \|x_t - x_{t+1}\| + \|y_t - y_{t+1}\| + G \sum_{t=1}^{T} \|y_t - y_{t+1}\| \\
&\leq G \sum_{t=1}^{T} \frac{4G}{Ht} + G \sum_{t=1}^{T} \frac{4G}{Ht} && \text{by Lemma 45}
\end{aligned}$$

$$\leq \frac{8G^2}{H}(1 + \int_1^T \frac{1}{t}dt)$$
$$= \frac{8G^2}{H}(1 + \ln T).$$

The other side of the inequality follows analogously by using the other inequality in Lemma 44. □

4.7 Training Generative Adversarial Networks

In this section we use our ideas to train Generative Adversarial Networks (GANs) [63].

4.7.1 GAN Formulation

GANs are particular approach to generative modeling. A *generative model* is is a machine learning model that takes samples drawn from an unknown distribution p_{data} and learns to represent an estimate of that distribution. After training, the model outputs a distribution p_{model} or some way to generate samples from it [62]. A GAN can be though of as two neural networks, the *generator* G and the *discriminator* D, playing a game against each other. The goal of the generator is to create samples from p_{model} that look like samples from p_{data}, and the goal of the discriminator is to recognize if a given sample comes from p_{data} or if it is a fake sample generated by its adversary. The original GAN formulation from [63] poses the problem as finding a solution to

$$\min_G \max_D \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[\log(D(\boldsymbol{x}))] + \mathbb{E}_{z\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]. \tag{4.15}$$

Here $p_{\boldsymbol{z}}(\boldsymbol{z})$ is some noise distribution that G maps onto the data space. Generative models have plenty of applications in other areas of machine learning, for example: reinforcement learning [52], semi-supervised learning [113, 123], single image super resolution [91], image-to-image translation [77], and even art creation [31], just to mention a few.

4.7.2 Mode Collapse

The most natural approach to train a GAN (and the original one used in [63]), is to simultaneously perform gradient descent on the generator's parameters and gradient ascent on those of the discriminator. However, it has been shown that even in simple convex-concave games such as $\mathcal{L}(x, y) = xy$, if one performs gradient descent on x and gradient ascent on y the dynamics do not necessarily converge to the Nash Equilibrium (see Ch. 5 of [62]). So it should not be surprising to observe that serious problems arise while training a GAN. We say a GAN suffered from mode collapse if the generator ends up producing samples from only a few modes from the distribution p_{data}, visually it means that the generator produces samples with low diversity. The first row in Figure 4.1 shows a clear example of this.

Since the introduction of GANs there has been an incredible effort from the machine learning community to understand why mode collapse occurs and how to fix it. In a very recent large-scale study [95], many GAN models were thoroughly tested to see if one outperformed the others. Their conclusion was "we did not find evidence that any of the tested algorithms consistently outperforms the non-saturating GAN introduced in [63]". The algorithms/models tested in the aforementioned study were: MM-GAN [63], NS-GAN [63], WGAN [12], WGAN GP [65], LS GAN [98], DRAGAN [85] and BEGAN [25].

The algorithm for training the non-saturating GAN from [63] corresponds to running two sublinear individual regret algorithms in parallel, one for the generator and another for the discriminator. However, it is common to observe mode collapse using this training procedure. In view of Theorem 21 we tested a variant of SP-RFTL on this setting hoping for significantly different training dynamics.

4.7.3 SP-RFTL for Training GANs

Even though the original GAN formulation, Equation 4.15, is not convex-concave we tested a variant of SP-RFTL on this setting. The particular implementation consists on 1) taking a mini-batch of data from p_{data} with N samples to approximate the payoff function in

Equation 4.15 with

$$\mathcal{L}_t(G, D) = \frac{1}{N}\sum_{n=1}^{N} \log(D(\boldsymbol{x_n})) + \mathbb{E}_{z\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))],$$

2) simultaneously run a sublinear individual Regret algorithm on the generator's parameters and another one on the discriminator's parameters for a fixed number of iterations and 3) uniformly average the iterates of both algorithms. Then we sample a new mini-batch of data from p_{data} to obtain $\mathcal{L}_{t+1}$ and repeat the procedure. If the payoff function $\mathcal{L}_t$ were convex-concave then the combination of steps 2) and 3) would be equivalent to finding an approximate NE for $\mathcal{L}_t$. It is easy to see that the procedure just described follows the spirit of SP-RFTL where both R_X, R_Y are set to any constant function. The reason for this is that the sequence $\mathcal{L}_t$ is stochastic (not adversarial) and thus regularization is not necessary.

4.7.4 Experiments

In Figure 4.1 we compare our proposed algorithm SP-RFTL with: Unrolled GAN [99], Wassertein GAN [12], and Wassertein GAN with Gradient Clipping [65]. The dataset is a mixture of eight gaussians placed uniformly in a circle of radius two with variance .02. The generator and discriminator architectures for Unrolled 0, Unrolled 4, and SP-RFTL are identical to those in Appendix A from [99]. The optimization parameters for Unrolled 0 and Unrolled 4 are the ones suggested in [99]. The optimization parameters for SP-RFTL are the same as for Unrolled 0, the extra parameter that controls how often we average the iterates was tuned by visual inspection. The WGAN and WGAN-GC architectures and parameters are exactly the ones provided in [65]. WGAN and WGAN-GC use an extra fully connected hidden layer compared to Unrolled 0, Unrolled 4, and SP-RFTL, we did not change the architecture assuming [65] did their best effort to produce their original results. All the algorithms use Adam [84, 110] as an optimizer. All the experiments were run on a Mac-Book Pro with processor 3.1 GHz Intel Core i7, and 16 GB

of RAM. In particular, no GPU was used. All the code for this project can be found at https://github.com/adrianriv/gans-mode-collapse.

We judge the performance of the generator based on the quality of its samples. From Figure 4.1 it is obvious that SP-RFTL learns the correct underlying distribution in the shortest amount of time. A final remark is that Unrolled 0 corresponds to running two no-individual regret algorithms in parallel, which results in mode collapse. Interestingly, SP-RFTL the algorithm with best performance, is doing exactly the same with the difference that it is averaging its iterates every fixed number of rounds.

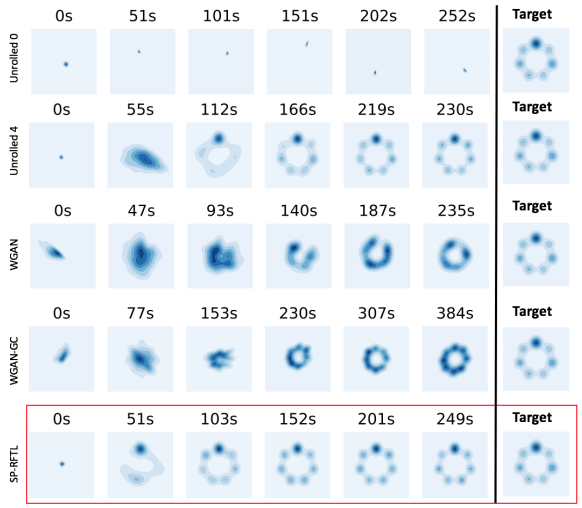

Figure 4.1: Comparison of algorithms in the mixture of 8 gaussians dataset. Each image shows the probability density produced by the generator after x seconds (CPU time) of training. It is clear that SP-RFTL (in red) outperforms all other algorithms.

4.8 Conclusion

In this chapter, we considered the Online Matrix Games problem, where two players interact in a sequence of zero-sum games with arbitrarily changing payoff matrices. The goal for both players is to achieve small Nash Equilibrium (NE) regret, that is, the players want to ensure their average payoffs over T rounds are close to those in the NE of the mean payoff matrix in hindsight. While it is known that standard Online Convex Optimization algorithms such as Online Gradient Descent can be used to find approximate equilibria in

static zero-sum games, our impossibility result shows that no algorithm for online convex optimization can achieve sublinear Nash Equilibrium regret ($o(T)$) when the sequence of payoffs are chosen arbitrarily. We then design and analyze algorithms that achieve sublinear NE regret for the Online Matrix Games problem, under both full information feedback and bandit feeback settings. In the full information case, the performance of the algorithm is optimal with respect to the number of rounds (up to logarithm factors) and depends logarithmically on the number of actions of each player. For the bandit feedback setting, we provide an algorithm with sublinear NE regret using a one-point matrix estimate. Lastly, we test our algorithm for training GANs on a basic setup and obtain satisfactory results.

CHAPTER 5
LARGE SCALE MARKOV DECISION PROCESSES WITH CHANGING REWARDS

5.1 Introduction

In this chapter, we study Markov Decision Processes (hereafter MDPs) with arbitrarily varying rewards. MDP provides a general mathematical framework for modeling sequential decision making under uncertainty [26, 75, 109]. In the standard MDP setting, if the process is in some state s, the decision maker takes an action a and receives an expected reward of $r(s, a)$. The process then randomly enters a new state according to some known transition probability.

In particular, the standard MDP model assumes that the decision maker has complete knowledge of the reward function $r(s, a)$, which does not change over time.

Over the past two decades, there has been much interest in sequential learning and decision making in an unknown and possibly *adversarial* environment. A wide range of sequential learning problems can be modeled using the framework of Online Convex Optimization (OCO) [72, 139]. In the OCO setting, the decision maker plays a repeated game against an adversary for a given number of rounds. At the beginning of each round indexed by t, the decision maker chooses an action a_t from a convex compact set A and the adversary chooses a concave reward function $r_t(\cdot)$, hence a reward of $r_t(a_t)$ is received. After observing the realized reward function, the decision maker chooses its next action a_{t+1} and so on. Since the decision maker does not know the future reward functions, its goal is to achieve a small *regret*; that is, the cumulative reward earned throughout the game should be close to the cumulative reward if the decision maker had been given the benefit of hindsight

to choose a fixed action. We can express the regret for T rounds as

$$\text{Regret}(T) = \max_{a \in A} \sum_{t=1}^{T} r_t(a) - \sum_{t=1}^{T} r_t(a_t).$$

The OCO model has many applications such as universal portfolios [42, 73, 82], online shortest path [124], and online submodular minimization [68]. It is also closely related with areas such as convex optimization [24, 69] and game theory [36]. There are many algorithms that guarantee sublinear regret, e.g., Online Gradient Descent [139], Perturbed Follow the Leader [83], and Regularized Follow the Leader [6, 116].

Compared with the MDP setting, the main difference is that in OCO there is no notion of states, however the payoffs may be chosen by an adversary.

In this work, we study a general problem framework that unites MDP and OCO, which we call the **Online MDP problem**. More specifically, we consider MDPs where the transition probabilities are known but the rewards are sequentially chosen by an adversary.

We list below some canonical motivating examples that can be modeled as Online MDPs.

- Adversarial Multi-Armed Bandits with Constraints [136]: We can generalize the adversarial multi-armed bandits problem with k arms (see Auer et al. [15]) with various constraints such as: restricting the number of times that an arm can be chosen in a given time interval, limiting how we switch between arms, etc. These constraints can be captured easily by defining appropriate states in the Online MDP.

- The Paging Problem [50]: Suppose we are given n pages. A memory can hold at most k ($k < n$) of them. An arbitrary sequence of paging request arrives. A page request is a *hit* if the associated page is in memory, and is a *miss* otherwise. After each request, the decision maker may swap any page in memory by paying some cost. Note that the state of the memory and the swapping decisions can be modeled using MDP. The decision maker's goal is to maximize the number of hits minus the

switching costs.

- The k-Server Problem [50, 86]: In this classical problem in computer science, there are k servers, represented as points in a metric space. Requests arrive to the metric space, which are also represented as points. As each request arrives, the decision maker can choose to move one of the servers to the requested point. The goal is to minimize the total distance all servers move.

  If the arrivals of requests are adversarial, this problem can be modeled as an Online MDP problem, where the state represents the position of servers.

Notice that in all of the problems above, the transition probabilities are known, while the adversarial rewards/costs are observed by the decision maker sequentially after each decision epoch. Moreover, in each of these Online MDP problems, the size of the state space may grow exponentially with the number k. Some other noteworthy examples are the stochastic inventory control problem [109] and some server queuing problems [2, 43].

5.1.1 Main Results

We propose a new computationally efficient algorithm that achieves near optimal regret for the Online MDP problem. Our algorithm is based on the (dual) linear programming formulation of infinite-horizon average reward MDPs, which uses the occupancy measure of state-action pairs as decision variables. This approach differs from other papers that have studied the Online MDP problem previously, see review in §5.1.2.

We prove that the algorithm's regret is bounded by $O(\tau + \sqrt{\tau T(\ln|S| + \ln|A|)}\ln(T))$, where S denotes the state space, A denotes the action space, τ is the mixing time of the MDP, and T is the number of periods. Notice that this regret bound depends *logarithmically* on the size of the state and action space. The algorithm solves a regularized linear program in each period with $poly(|S||A|)$ complexity. The regret bound and the computation complexity compares favorably to the existing methods, which are summarized in

§5.1.2.

We then extend our results to the case where the state space S is extremely large so that $poly(|S||A|)$ computational complexity is impractical. We assume the state-action occupancy measures associated with stationary policies are approximated with a linear architecture of dimension $d \ll |S|$.

We design an approximate algorithm combining several innovative techniques for solving large scale MDPs inspired by [2, 3]. A salient feature of this algorithm is that its computational complexity does not depend on the size of the state-space but instead on the number of features d. The algorithm has a regret bound $O(c_{S,A}(\ln|S| + \ln|A|)\sqrt{\tau T}\ln T)$, where $c_{S,A}$ is a problem dependent constant. To the best of our knowledge, this is the first $\tilde{O}(\sqrt{T})$ regret result for large scale Online MDPs.

5.1.2 Related Work

The history of MDP goes back to the seminal work of Bellman [22] and Howard [75] from the 1950's.

Some classic algorithms for solving MDP include policy iteration, value iteration, policy gradient, Q-learning and their approximate versions (see [26, 27, 109] for an excellent discussion). In this work, we will focus on a relatively less used approach, which is based on finding the *occupancy measure* using linear programming, as done recently in [3, 40, 131] to solve MDPs with *static* rewards (see more details in Section 5.3.1). To deal with the curse of dimensionality, Chen et al. [40] uses bilinear functions to approximate the occupancy measures and Abbasi-Yadkori et al. [3] uses a linear approximation.

The Online MDP problem was first studied a decade ago by [50, 136]. Even-Dar et al. [50] developed no regret algorithms where the bound scales as $O(\tau^2\sqrt{T\ln(|A|)})$, where τ is the mixing time defined in §5.2. Their method runs an expert algorithm (e.g. Weighted Majority [93]) on every state where the actions are the experts. However, the authors did not consider the case with large state space in their paper.

Yu et al. [136] proposed a more computationally efficient algorithm using a variant of Follow the Perturbed Leader [83], but unfortunately their regret bound becomes $O(|S||A|^2\tau T^{3/4+\epsilon})$. They also considered approximation algorithm for large state space, but did not establish an exact regret bound. The work most closely related to ours is that from Dick et al. [44], where the authors also use a linear programming formulation of MDP similar to ours.

However, there seem to be some gaps in the proof of their results.[1] That issue aside, in order to solve large-scale MDPs, their focus is to efficiently solve the quadratic sub-problems that define their iterates efficiently. Instead, we leverage the linear approximation scheme introduced in [3].

Ma et al. [96] also considers Online MDPs with large state space. Under some conditions, they show sublinear regret using a variant of approximate policy iteration, but the regret rate is left unspecified in their paper. Zimin and Neu [138] considered a special class of MDPs called *episodic* MDPs and design algorithms using the occupancy measure LP formulation. Following this line of work, Neu et al. [106] shows that several reinforcement learning algorithms can be viewed as variant of Mirror Descent [80], thus one can establish convergence properties of these algorithms. In [105], the authors considered Online MDPs with bandit feedback and provide an algorithm based on [50]'s with regret of $O(T^{2/3})$. Some other related work can be found in [39, 81, 88].

A more general problem to the Online MDP setting considered here is where the MDP transition probabilities also change in an adversarial manner, which is beyond the scope of this chapter. It is believed that this problem is much less tractable computationally; see discussion in [48]. Yu and Mannor [134] studied MDPs with changing transition probabilities, although [105] questions the correctness of their result, as the regret obtained seems to have broken a lower bound. In [59], the authors use a sliding window approach under

[1] In particular, we believe the proof of Lemma 1 in [44] is incorrect. Equation (8) in their paper states that the regret relative to a policy is equal to the sum of a sequence of vector products; however, the dimensions of vectors involved in these dot products are incompatible. By their definition, the variable ν_t is a vector of dimension $|S|$, which is being multiplied with a loss vector with dimension $|S||A|$.

a particular definition of regret. Abbasi-Yadkori et al. [1] achieved sublinear regret with changing transition probabilities when compared against a restricted policy class.

5.2 Problem Formulation: Online MDP

We consider a general Markov Decision Process (MDP) with known transition probabilities but unknown and adversarially chosen rewards. Let S denote the set of possible states, and A denote the set of actions. (For notational simplicity, we assume the set of actions a player can take is the same for all states, but this assumption can be relaxed easily.) At each period $t \in [T]$, if the system is in state $s_t \in S$, the decision maker chooses an action $a_t \in A$ and collects a reward $r_t(s_t, a_t)$. Here, $r_t : S \times A \to [-1, 1]$ denotes a reward function for period t. We assume that the sequence of reward functions $\{r_t\}_{t=1}^T$ is initially unknown to the decision maker. The function r_t is revealed only after the action a_t has been chosen. We allow the sequence $\{r_t\}_{t=1}^T$ to be chosen by an *adaptive adversary*, meaning r_t can be chosen using the history $\{s_i\}_{i=1}^t$ and $\{a_i\}_{i=1}^{t-1}$. In particular, the adversary does not observe the action a_t when choosing r_t. After a_t is chosen, the system then proceeds to state s_{t+1} in the next period with probability $P(s_{t+1}|s_t, a_t)$. We assume the decision maker has complete knowledge of the transition probabilities given by $P(s'|s, a) : S \times A \to S$.

Suppose that the initial state of the MDP follows $s_1 \sim \nu_1$, where ν_1 is a probability distribution over S. The objective of the decision maker is to choose a sequence of actions based on the history of states and rewards observed, such that the cumulative reward in T periods is close to that of the optimal offline static policy. Formally, let π denote a stationary (possibly randomized) policy: $\pi : S \to \Delta_A$, where Δ_A is the set of probability distributions over the action set A. Let Π denote the set of all stationary policies. We aim to find an algorithm that minimizes

$$\text{MDP-Regret}(T) \triangleq \sup_{\pi \in \Pi} R(T, \pi), \text{ with } R(T, \pi) \triangleq \mathbb{E}[\sum_{t=1}^{T} r_t(s_t^\pi, a_t^\pi)] - \mathbb{E}[\sum_{t=1}^{T} r_t(s_t, a_t)], \tag{5.1}$$

where the expectations are taken with respect to random transitions of MDP and (possibly) external randomization of the algorithm.

5.3 Preliminaries

Next, we provide additional notations for the MDP. Let $P^{\pi}_{s,s'} \triangleq P(s' \mid s, \pi(s))$ be the probability of transitioning from state s to s' given a policy π. Let P^{π} be an $|S| \times |S|$ matrix with entries $P^{\pi}_{s,s'}$ $(\forall s, s' \in S)$. We use row vector $\nu_t \in \Delta_S$ to denote the probability distribution over states at time t. Let ν^{π}_{t+1} be the distribution over states at time $t+1$ under policy π, given by $\nu^{\pi}_{t+1} = \nu_t P^{\pi}$. Let ν^{π}_{st} denote the stationary distribution for policy π, which satisfies the linear equation $\nu^{\pi}_{st} = \nu^{\pi}_{st} P^{\pi}$. We assume the following condition on the convergence to stationary distribution, which is commonly used in the MDP literature [see 50, 105, 136].

Assumption 1. *There exists a real number $\tau \geq 0$ such that for any policy $\pi \in \Pi$ and any pair of distributions $\nu, \nu' \in \Delta_S$, it holds that $\|\nu P^{\pi} - \nu' P^{\pi}\|_1 \leq e^{-\frac{1}{\tau}} \|\nu - \nu'\|_1$.*

We refer to τ in Assumption 1 as the *mixing time*, which measures the convergence speed to the stationary distribution. In particular, the assumption implies that ν^{π}_{st} is unique for a given policy π.

We use $\mu(s, a)$ to denote the proportion of time that the MDP visits state-action pair (s, a) in the long run. We call $\mu^{\pi} \in \mathbb{R}^{|S| \times |A|}$ the *occupancy measure* of policy π. Let ρ^{π}_t be the long-run average reward under policy π when the reward function is fixed to be r_t every period, i.e., $\rho^{\pi}_t \triangleq \lim_{T \to \infty} \frac{1}{T} \sum_{i=1}^{T} \mathbb{E}[r_t(s^{\pi}_i, a^{\pi}_i)]$. We define $\rho_t \triangleq \rho^{\pi_t}_t$, where π_t is the policy selected by the decision maker at time t.

5.3.1 Linear Programming Formulation for the Average Reward MDP

Given a reward function $r : S \times A \to [-1, 1]$, suppose one wants to find a policy π that maximizes the long-run average reward: $\rho^* = \sup_{\pi} \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} r(s^{\pi}_t, a^{\pi}_t)$.

Under Assumption 1, the Markov chain induced by any policy is ergodic and the long-run average reward is independent of the starting state (see [26]). It is well known that the optimal policy can be obtained by solving the Bellman equation, which in turn can be written as a linear program (in the dual form):

$$\begin{aligned} \rho^* = \max_{\mu} & \sum_{s\in S}\sum_{a\in A} \mu(s,a) r(s,a) \\ \text{s.t.} & \sum_{s\in S}\sum_{a\in A} \mu(s,a) P(s'|s,a) = \sum_{a\in A} \mu(s',a) \quad \forall s' \in S \\ & \sum_{s\in S}\sum_{a\in A} \mu(s,a) = 1, \quad \mu(s,a) \geq 0 \quad \forall s \in S, \forall a \in A. \end{aligned} \tag{5.2}$$

Let μ^* be an optimal solution to the LP (5.2). We can construct an optimal policy of the MDP by defining $\pi^*(s,a) \triangleq \frac{\mu^*(s,a)}{\sum_{a\in A}\mu^*(s,a)}$ for all $s \in S$ such that $\sum_{a\in A}\mu^*(s,a) > 0$; for states where the denominator is zero, the policy may choose arbitrary actions, since those states will not be visited in the stationary distribution. Let ν^*_{st} be the stationary distribution over states under this optimal policy.

For simplicity, we will write the first constraint of LP (5.2) in the matrix form as $\mu^\top(P - B) = 0$, where B is an appropriately chosen matrix with 0-1 entries. We denote the feasible set of the above LP as $\Delta_M \triangleq \{\mu \in \mathbb{R} : \mu \geq 0, \mu^\top 1 = 1, \mu^\top(P-B) = 0\}$. The following definition will be used in the analysis later.

Definition 6. *Let $\delta_0 \geq 0$ be the largest real number such that for all $\delta \in [0, \delta_0]$, the set $\Delta_{M,\delta} \triangleq \{\mu \in \mathbb{R}^{|S|\times|A|} : \mu \geq \delta, \mu^\top 1 = 1, \mu^\top(P-B) = 0\}$ is nonempty.*

5.4 A Sublinear Regret Algorithm for Online MDP

In this section, we present an algorithm for the Online MDP problem. The algorithm is very intuitive given the LP formulation (5.2) for the static problem. As the rewards may change each round, the algorithm simply treats the Online MDP problem as an Online Convex

Optimization (OCO) problem with reward functions $\{r_t\}_{t=1}^T$ and decision set Δ_M.

Algorithm 13 (MDP-RFTL)

input: parameter $\delta > 0, \eta > 0$, regularization term $R(\mu) = \sum_{s\in S}\sum_{a\in A}\mu(s,a)\ln(\mu(s,a))$
initialization: choose any $\mu_1 \in \Delta_{M,\delta} \subset \mathbb{R}^{|S|\times|A|}$
for $t = 1, ...T$ **do**
  observe current state s_t
  if $\sum_{a\in A}\mu_t(s_t,a) > 0$ **then**
    choose action $a \in A$ with probability $\frac{\mu_t(s_t,a)}{\sum_a \mu_t(s_t,a)}$.
  else
    choose action $a \in A$ with probability $\frac{1}{|A|}$
  end if
  observe reward function $r_t \in [-1,1]^{|S||A|}$
  update $\mu_{t+1} \leftarrow \arg\max_{\mu\in\Delta_{M,\delta}} \sum_{i=1}^t \left[\langle r_i, \mu\rangle - \frac{1}{\eta}R(\mu)\right]$
end for

At the beginning of each round $t \in [T]$, the algorithm starts with an occupancy measure μ_t. If the MDP is in state s_t, we play action $a \in A$ with probability $\frac{\mu_t(s_t,a)}{\sum_a \mu_t(s_t,a)}$. If the denominator is 0, the algorithm picks any action in A with equal probability. After observing reward function r_t and collecting reward $r_t(s_t, a_t)$, the algorithm changes the occupancy measure to μ_{t+1}.

The new occupancy measure is chosen according to the Regularized Follow the Leader (RFTL) algorithm [6, 116]. RFTL chooses the best occupancy measure for the cumulative reward observed so far, $\sum_{i=1}^t r_i$, plus a regularization term $R(\mu)$. The regularization term forces the algorithm not to drastically change the occupancy measure from round to round. In particular, we choose $R(\mu)$ to be the entropy function. This choice will allow us to get $\ln(|S||A|)$ dependence in the regret bound.

The complete algorithm is shown in Algorithm 13. The main result of this section is the following.

Theorem 27. *Suppose $\{r_t\}_{t=1}^T$ is an arbitrary sequence of rewards such that $|r_t(s,a)| \le 1$ for all $s \in S$ and $a \in A$. For $T \ge \ln^2(1/\delta_0)$, the MDP-RFTL algorithm with parameters*

$\eta = \sqrt{\frac{T \ln(|S||A|)}{\tau}}$, $\delta = e^{-\sqrt{T}/\sqrt{\tau}}$ *guarantees*

$$\textit{MDP-Regret}(T) \leq O\left(\tau + 4\sqrt{\tau T(\ln|S| + \ln|A|)\ln(T)}\right).$$

The regret bound in Theorem 27 is near optimal: a lower bound of $\Omega(\sqrt{T \ln|A|})$ exists for the problem of learning with expert advice [57, 72], a special case of Online MDP where the state space is a singleton. We note that the bound only depends *logarithmically* on the size of the state space and action space.

The state-of-the-art regret bound for Online MDPs is that of [50], which is $O(\tau + \tau^2\sqrt{\ln(|A|)T})$. Compared to their result, our bound is better by a factor of $\tau^{3/2}$. However, our bound has depends on $\sqrt{\ln|S| + \ln|A|}$, whereas the bound in [50] depends on $\sqrt{\ln|A|}$. Both algorithms require $poly(|S||A|)$ computation time, but are based on different ideas: the algorithm of [50] is based on expert algorithms and requires computing Q-functions at each time step, whereas our algorithm is based on RFTL. In the next section, we will show how to extend our algorithm to the case with large state space.

5.4.1 Sketch of Analysis of MDP-RFTL

The key to analyze our algorithm is to decompose the regret with respect to policy $\pi \in \Pi$ as follows

$$R(T,\pi) = \left[\mathbb{E}[\sum_{t=1}^{T} r_t(s_t^\pi, a_t^\pi)] - \sum_{t=1}^{T}\rho_t^\pi\right] + \left[\sum_{t=1}^{T}\rho_t^\pi - \sum_{t=1}^{T}\rho_t\right] + \left[\sum_{t=1}^{T}\rho_t - \mathbb{E}[\sum_{t=1}^{T} r_t(s_t, a_t)]\right]. \tag{5.3}$$

This decomposition was first used by [50]. We now give some intuition on why $R(T,\pi)$ should be sublinear. By the mixing condition in Assumption 1, the state distribution ν_t^π at time t under a policy π differs from the stationary distribution ν_{st}^π by at most $O(\tau)$. This result can be used to bound the first term of (5.3).

The second term of (5.3) can be related to the online convex optimization (OCO) prob-

lem through the linear programming formulation from §5.3.1. Notice that $\rho_t^\pi = \sum_{s\in S}\sum_{a\in A}\mu^\pi(s,a)r(s,a) = \langle \mu^\pi, r\rangle$, and $\rho_t = \sum_{s\in S}\sum_{a\in A}\mu_t^\pi(s,a)r(s,a) = \langle \mu^{\pi_t}, r\rangle$. Therefore, we have

$$\sum_{t=1}^{T}\rho_t^\pi - \sum_{t=1}^{T}\rho_t = \sum_{t=1}^{T}\langle \mu^\pi, r_t\rangle - \sum_{t=1}^{T}\langle \mu^{\pi_t}, r_t\rangle, \tag{5.4}$$

which is exactly the regret quantity commonly studied in the OCO problem. We are thus seeking an algorithm that can bound $\max_{\mu^\pi\in\Delta_M}\sum_{t=1}^{T}\langle \mu^\pi, r_t\rangle - \sum_{t=1}^{T}\langle \mu^{\pi_t}, r_t\rangle$. In order to achieve logarithmic dependence on $|S|$ and $|A|$ in Theorem 27, we apply the RFTL algorithm, regularized by the negative entropy function $R(\mu)$. A technical challenge we faced in the analysis is that $R(\mu)$ is not Lipschitz continuous over the feasible set Δ_M. So we design the algorithm to play in a shrunk set $\Delta_{M,\delta}$ for some $\delta > 0$ (see Definition 6), in which $R(\mu)$ is indeed Lipschitz continuous.

For the last term in (5.3), note that it is similar to the first term, albeit more complicated: the policy π is fixed in the first term, but the policy π_t used by the algorithm is varying over time. To solve this challenge, the key idea is to show that the policies do not change too much from round to round, so that the third term grows sublinearly in T. To this end, we use the property of the RFTL algorithm with a carefully chosen regularization parameter $\eta > 0$. The complete proof of Theorem 27 can be found in Section 5.5.

5.5 Regret Analysis of MDP-RFTL

To bound the regret incurred by MDP-RFTL, we bound each term in Eq (5.3). We start with the first term. We use the following lemma, which was first stated in [50] and was also used by [105].

Lemma 46. *For any $T \geq 1$ and any policy π it holds that*

$$\mathbb{E}[\sum_{t=1}^{T} r_t(s_t^\pi, a_t^\pi)] - \sum_{t=1}^{T}\rho_t^\pi \leq 2\tau + 2.$$

Proof of Lemma 46. Recall that $|r_t(s,a)| \leq 1$, so we have $|\sum_{a\in A}\pi(s,a)r_t(s,a)| \leq 1$ by Cauchy-Schwarz inequality, since $\pi(s,\cdot)$ defines a probability distribution over actions. Also, recall that ν_t^π is the stationary distribution over states by following policy π and $\nu_{t+1}^\pi = \nu_t^\pi P^\pi$ for all $t \in [T]$. We have

$$\begin{aligned}
\mathbb{E}[\sum_{t=1}^{T} r_t(s_t^\pi, a_t^\pi)] - \sum_{t=1}^{T}\rho_t^\pi &= \sum_{t=1}^{T}\sum_{s\in S}(\nu_t^\pi(s) - \nu_{st}^\pi(s))\sum_{a\in A}\pi(s,a)r_t(s,a) \\
&\leq \sum_{t=1}^{T}\sum_{s\in S}\nu_t^\pi(s) - \nu_{st}^\pi(s) \\
&\leq \sum_{t=1}^{T}\|\nu_t^\pi(s) - \nu_{st}^\pi(s)\|_1.
\end{aligned}$$

Now, notice that

$$\begin{aligned}
\|\nu_t^\pi(s) - \nu_{st}^\pi(s)\|_1 &= \|\nu_{t-1}^\pi P^\pi - \nu_{st}P^\pi\|_1 \\
&\leq e^{-\frac{1}{\tau}}\|\nu_{t-1}^\pi - \nu_{st}\|_1 \quad \text{by Assumption 1} \\
&\leq e^{-\frac{t}{\tau}}\|\nu_1^\pi - \nu_{st}^\pi\|_1 \quad \text{by repeating the argument } t-1 \text{ more times} \\
&\leq 2e^{-\frac{t}{\tau}}.
\end{aligned}$$

Finally, we have that

$$\begin{aligned}
\sum_{t=1}^{T}\|\nu_t^\pi(s) - \nu_{st}^\pi(s)\|_1 &\leq 2\sum_{t=1}^{T} e^{-\frac{t}{\tau}} \\
&\leq 2(1 + \int_0^\infty e^{-\frac{t}{\tau}})dt \\
&= 2\tau + 2,
\end{aligned}$$

which concludes the proof. □

We now bound the third term in (5.3). We use the following lemma, which bounds the difference of two stationary distributions by the difference of the corresponding occupancy

measures.

Lemma 47. *Let ν^1_{st} and ν^2_{st} be two arbitrary stationary distributions over S. Let μ^1 and μ^2 be the corresponding occupancy mesures. It holds that*

$$\|\nu^1_{st} - \nu^2_{st}\|_1 \leq \|\mu^1 - \mu^2\|_1.$$

Proof of Lemma 47.

$$\begin{aligned}
\|\nu^1_{st} - \nu^2_{st}\|_1 &= \sum_{s\in S} |\nu^1_{st}(s) - \nu^2_{st}(s)| \\
&= \sum_{s\in S} |\sum_{a\in A} \mu^1(s,a) - \mu^2(s,a)| \\
&\leq \sum_{s\in S}\sum_{a\in A} |\mu^1(s,a) - \mu^2(s,a)| \\
&= \|\mu^1 - \mu^2\|_1.
\end{aligned}$$

□

We are ready to bound the third term in (5.3).

Lemma 48. *Let $\{s_t, a_t\}_{t=1}^T$ be the random sequence of state-action pairs generated by the policies induced by occupancy measures $\{\mu^{\pi_t}\}_{t=1}^T$. It holds that*

$$\sum_{t=1}^T \rho_t - \mathbb{E}\left[\sum_{t=1}^T r_t(s_t, a_t)\right] \leq \sum_{t=1}^T 2e^{-\frac{t-1}{\tau}} + \sum_{t=1}^T \sum_{\theta=0}^{t-1} e^{-\frac{\theta}{\tau}} \|\mu^{\pi_{t-\theta}} - \mu^{\pi_{t-(\theta+1)}}\|_1.$$

Proof of Lemma 48. By the definition of ρ_t, we have

$$\begin{aligned}
\sum_{t=1}^T \rho_t - \mathbb{E}\left[\sum_{t=1}^T r_t(s_t, a_t)\right] &= \sum_{t=1}^T \sum_{s\in S} (\nu^{\pi_t}_{st}(s) - \nu^t(s)) \sum_{a\in A} \pi^t(s,a) r_t(s,a) \\
&\leq \sum_{t=1}^T \|\nu^{\pi_t}_{st} - \nu^t\|_1.
\end{aligned}$$

Now, recall that $\nu_t = \nu_1 P^{\pi_1} P^{\pi_2} ... P^{\pi_{t-1}}$. We now bound $\|\nu_{st}^{\pi_t} - \nu^t\|_1$ for all $t \in [T]$ as follows:

$$\begin{aligned}
\|\nu^t - \nu_{st}^{\pi_t}\|_1 &\leq \|\nu^t - \nu_{st}^{\pi_{t-1}}\|_1 + \|\nu_{st}^{\pi_{t-1}} - \nu_{st}^{\pi_t}\|_1 \\
&\leq \|\nu^t - \nu_{st}^{\pi_{t-1}}\|_1 + \|\mu^{\pi_{t-1}} - \mu^{\pi_t}\|_1 \quad \text{by Lemma 47} \\
&= \|\nu^{t-1} P^{\pi_{t-1}} - \nu_{st}^{\pi_{t-1}} P^{\pi_{t-1}}\|_1 + \|\mu^{\pi_{t-1}} - \mu^{\pi_t}\|_1 \\
&\leq e^{-\frac{1}{\tau}} \|\nu^{t-1} - \nu_{st}^{\pi_{t-1}}\|_1 + \|\mu^{\pi_{t-1}} - \mu^{\pi_t}\|_1 \quad \text{by Assumption 1} \\
&\leq e^{-\frac{1}{\tau}} (e^{-\frac{1}{\tau}} \|\nu^{t-2} - \nu_{st}^{\pi_{t-2}}\|_1 + \|\mu^{\pi_{t-2}} - \mu^{\pi_{t-1}}\|_1) + \|\mu^{\pi_{t-1}} - \mu^{\pi_t}\|_1 \\
&\leq e^{-\frac{t-1}{\tau}} \|\nu^1 - \nu_{st}^{\pi_1}\|_1 + \sum_{\theta=0}^{t-1} e^{-\frac{\theta}{\tau}} \|\mu^{\pi_{t-\theta}} - \mu^{\pi_{t-(\theta+1)}}\|_1,
\end{aligned}$$

which yields the desired claim. □

Combining Lemma 46, Lemma 48 and Eq (5.3), we have arrived at the following bound on the regret:

$$\begin{aligned}
R(T, \pi) \leq (2\tau{+}2){+}&\left[\sum_{t=1}^{T} \langle \mu^{\pi}, r_t \rangle {-} \sum_{t=1}^{T} \langle \mu^{\pi_t}, r_t \rangle\right] \\
&+\left[\sum_{t=1}^{T} 2e^{-\frac{t-1}{\tau}} {+} \sum_{t=1}^{T} \sum_{\theta=0}^{t-1} e^{-\frac{\theta}{\tau}} \|\mu^{\pi_{t-\theta}} {-} \mu^{\pi_{t-(\theta+1)}}\|_1\right].
\end{aligned}$$

To complete the proof, we want to bound the second and the third terms. For the second term $\max_{\mu \in \Delta_M} \sum_{t=1}^{T} \langle \mu^{\pi}, r_t \rangle - \sum_{t=1}^{T} \langle \mu^{\pi_t}, r_t \rangle$, since the reward functions are linear in μ and the set Δ_M is convex, any algorithm for Online Linear Optimization, e.g., Online Gradient Ascent [139], ensures a regret bound that is sublinear T. However, this would yield an MDP-regret rate that depends linearly on $|S| \times |A|$.

Instead, by noticing that the feasible set of the LP, Δ_M, is a subset of the probability simplex $\Delta^{|S||A|}$, we use RFTL and regularize using the negative entropy function. This will give us a rate that scales as $\ln(|S||A|)$, which is much more desirable than $O(|S||A|)$. Notice that the algorithm does not work with the set Δ_M directly but with $\Delta_{M,\delta}$ instead,

this is because the negative entropy is not Lipschitz over Δ_M. Working over $\Delta_{M,\delta}$ is the key to being able to bound the third term in the regret decomposition. Formally, we have the following result.

Lemma 49. *Let $\{\mu_t\}_{t=1}^T$ be the iterates of MDP-RFTL, then it holds that*

$$\max_{\mu\in\Delta_{M,\delta}} \sum_{t=1}^{T} \langle r_t, \mu\rangle \leq \sum_{t=1}^{T} \langle r_t, \mu^{\pi_{t+1}}\rangle + \frac{T}{\eta} \max_{\mu_1,\mu_2\in\Delta_{M,\delta}} \left[R(\mu_1) - R(\mu_2)\right].$$

Proof of Lemma 49. Define $f_t \triangleq \langle \mu, r_t\rangle$ and $f_t^R \triangleq f_t(\mu) - \frac{1}{\eta}R(\mu)$ for all $t = 1, .., T$. We first prove by induction that

$$\max_{\mu\in\Delta_{M,\delta}} \sum_{t=1}^{T} f_t^R(\mu) \leq \sum_{t=1}^{T} f_t^R(\mu^{\pi_{t+1}}).$$

The base case $T = 1$ is trivial by the definition of μ^{π_2}. Suppose the claim holds for $T-1$. For all $\mu \in \Delta_{M,\delta}$, we have that

$$\begin{aligned}
\sum_{t=1}^{T} f_t^R(\mu) &\leq \sum_{t=1}^{T} f_t^R(\mu^{\pi_{T+1}}) \\
&\leq \max_{\mu\in\Delta_{M,\delta}} \sum_{t=1}^{T-1} f_t^R(\mu) + f_T^R(\mu^{\pi_{T+1}}) \\
&\leq \sum_{t=1}^{T-1} f_t^R(\mu^{\pi_{t+1}}) + f_T^R(\mu^{\pi_{T+1}}) \quad \text{by induction hyposthesis} \\
&= \sum_{t=1}^{T} f_t^R(\mu^{\pi_{t+1}}).
\end{aligned}$$

The lemma follows by plugging back in the definition of f_t^R and rearranging terms. □

Lemma 50. *Let $\{\mu_t\}_{t=1}^T$ be the iterates of MDP-RFTL, it holds that*

$$\|\mu^{\pi_t} - \mu^{\pi_{t+1}}\|_1 \leq \frac{2\eta}{t}\left(1 + \frac{1}{\eta}G_R\right).$$

Proof of Lemma 50. Let $J(\mu) = \sum_{\theta=1}^{t} \left[\langle \mu, r_t \rangle - \frac{1}{\eta} R(\mu) \right]$. Since R is the negative entropy we know it is 1- strongly convex with respect to norm $\|\cdot\|_1$, thus J is $\frac{t}{\eta}$-strongly concave. By strong concavity we have

$$\frac{t}{2\eta} \|\mu^{\pi_{t+1}} - \mu^{\pi_t}\|_1^2 \leq J(\mu^{\pi_{t+1}}) - J(\mu^{\pi_t}) + \langle \nabla_\mu J(\mu^{\pi_{t+1}}), \mu^{\pi_t} - \mu^{\pi_{t+1}} \rangle.$$

Since $\mu^{\pi_{t+1}}$ is the optimizer of J the optimality condition states that $\langle \nabla_\mu J(\mu^{\pi_{t+1}}), \mu^{\pi_t} - \mu^{\pi_{t+1}} \rangle \leq 0$. Plugging back in the definition of J we have that

$$\begin{aligned}
&\frac{t}{2\eta} \|\mu^{\pi_{t+1}} - \mu^{\pi_t}\|_1^2 \\
&\leq \sum_{\theta=1}^{t} \left[\langle r_\theta, \mu^{\pi_{t+1}} \rangle - \frac{1}{\eta} R(\mu^{\pi_{t+1}}) \right] - \sum_{\theta=1}^{t} \left[\langle r_\theta, \mu^{\pi_t} \rangle - \frac{1}{\eta} R(\mu^{\pi_t}) \right] \\
&= \sum_{\theta=1}^{t-1} \left[\langle r_\theta, \mu^{\pi_{t+1}} \rangle - \frac{1}{\eta} R(\mu^{\pi_{t+1}}) \right] - \sum_{\theta=1}^{t-1} \left[\langle r_\theta, \mu^{\pi_t} \rangle - \frac{1}{\eta} R(\mu^{\pi_t}) \right] \\
&\qquad + \langle r_t, \mu^{\pi_{t+1}} \rangle - \frac{1}{\eta} R(\mu^{\pi_{t+1}}) - \langle r_\theta, \mu^{\pi_t} \rangle + \frac{1}{\eta} R(\mu^{\pi_t}) \\
&\leq \langle r_t, \mu^{\pi_{t+1}} \rangle - \frac{1}{\eta} R(\mu^{\pi_{t+1}}) - \langle r_\theta, \mu^{\pi_t} \rangle + \frac{1}{\eta} R(\mu^{\pi_t}) \quad \text{by definition of } \mu^{\pi_t} \\
&\leq \|r_t\|_\infty \|\mu^{\pi_t} - \mu^{\pi_{t+1}}\|_1 + \frac{1}{\eta} R(\mu^{\pi_t}) - \frac{1}{\eta} R(\mu^{\pi_{t+1}}) \quad \text{by Cauchy-Schwarz inequality} \\
&\leq \|\mu^{\pi_t} - \mu^{\pi_{t+1}}\|_1 + \frac{G_R}{\eta} \|\mu^{\pi_t} - \mu^{\pi_{t+1}}\|_1 \quad \text{Since } R \text{ is } G_R\text{- Lipschitz.}
\end{aligned}$$

By rearranging terms, we get

$$\|\mu^{\pi_t} - \mu^{\pi_{t+1}}\|_1 \leq \frac{2\eta}{t} \left(1 + \frac{1}{\eta} G_R \right).$$

□

Notice that by Lemma 50 we will need the regularizer R to be Lipschitz continuous with respect to norm $\|\cdot\|_1$. Unfortunately, the negative entropy function is not Lipschitz continuous over Δ_M, so we will force the algorithm to play in a shrunk set $\Delta_{M,\delta}$.

Lemma 51. *Let* $\Delta_\delta \triangleq \{x \in \mathbb{R}^d : \|x\|_1 = 1, x_i \geq \delta \ \forall i = 1, ..., d\}$. *The function* $R(x) \triangleq \sum_{i=1}^d x_i \ln(x_i)$ *is* G_R*-Lipschitz continuous with respect to* $\|\cdot\|_1$ *over* Δ_δ *with* $G_R = \max\{|\ln(\delta)|, 1\}$.

Proof of Lemma 51. We want to find $G_R > 0$ such that $\|\nabla R(x)\|_\infty \leq G_R$ for all $x \in \Delta_\delta$. Notice that $[\nabla R(x)]_i = 1 + \ln(x_i)$ for $i = 1, ...d$. Moreover, since for every $i = 1, ..., d$ we have $\delta \leq x_i \leq 1$ the following sequence of inequalities hold: $\ln(\delta) \leq 1 + \ln(\delta) \leq 1 + \ln(x_i) \leq 1$. It follows that $G_R = \max\{|\ln(\delta)|, 1\}$. □

The next lemma quantifies the loss in the regret due to playing in the shrunk set.

Lemma 52. *It holds that*

$$\max_{\mu \in \Delta_M} \sum_{t=1}^T \langle r_t, \mu \rangle \leq \max_{\mu \in \Delta_{M,\delta}} \sum_{t=1}^T \langle r_t, \mu \rangle + 2\delta T \left(|S||A| - 1\right).$$

Proof of Lemma 52. Given $z^* \in \Delta \subset \mathbb{R}^d$, define $z_p^* \triangleq \arg\min_{z \in \Delta_\delta} \|z - z^*\|_1$, with $\delta \leq \frac{1}{d}$. It holds that $\|z_p^* - z^*\|_1 \leq 2\delta(d-1)$. To see why the previous is true, choose $z^* = [1; 0; 0; ...; 0; 0]$. It is easily verified that $z_p^* = [1 - \delta(d-1); \delta; \delta; ...; \delta, \delta]$ and $\|z^* - z_p^*\|_1 = 2\delta(d-1)$. Because of the previous argument, if $\mu^* \in \arg\max_{\mu \in \Delta_M} \sum_{t=1}^T \langle r_t, \mu \rangle$ and μ_p^* is its $\|\cdot\|_1$ projection onto the set $\Delta_{M,\delta}$ then $\|\mu^* - \mu_p^*\|_1 \leq 2\delta(|S||A| - 1)$. The claim then follows since each function $\langle r_t, \mu \rangle$ is 1-Lipschitz continuous with respect to $\|\cdot\|_1$.

□

Given that we know the iterates of MDP-RFTL are close by Lemma 50, we can bound the last term in our regret bound

Lemma 53. *It holds that*

$$\sum_{t=1}^T 2e^{-\frac{t-1}{\tau}} + \sum_{t=1}^T \sum_{\theta=0}^{t-1} e^{-\frac{\theta}{\tau}} \|\mu^{\pi_{t-\theta}} - \mu^{\pi_{t-(\theta+1)}}\|_1 \leq 2(1+\tau) + 2\eta \left(1 + \frac{1}{\eta} G_R\right) (1 + \ln(T))(1+\tau).$$

Proof of Lemma 53. We first bound the first term

$$\sum_{t=1}^{T} 2e^{-\frac{t-1}{\tau}} \leq 2(1+\int_{1}^{\infty} e^{-\frac{x-1}{\tau}}dx) \leq 2(1+\tau).$$

We now bound the second term, let $\alpha = 2\eta\left(1+\frac{1}{\eta}G_R\right)$. We have that

$$\begin{aligned}
\sum_{t=1}^{T}\sum_{\theta=0}^{t-1} e^{-\frac{\theta}{\tau}}\|\mu^{\pi_{t-\theta}} - \mu^{\pi_{t-(\theta+1)}}\|_1 &= \alpha \sum_{t=1}^{T}\sum_{\theta=0}^{t-1} e^{-\frac{\theta}{\tau}}\frac{1}{t-\theta} \\
&= \alpha\left[e^{-\frac{0}{\tau}}\sum_{t=1}^{T} 1/t + e^{-\frac{1}{\tau}}\sum_{t=1}^{T-1} 1/t + e^{-\frac{2}{\tau}}\sum_{t=1}^{T-2} 1/t + ...\right] \\
&\leq \alpha\left[e^{-\frac{0}{\tau}}\sum_{t=1}^{T} 1/t + e^{-\frac{1}{\tau}}\sum_{t=1}^{T} 1/t + e^{-\frac{2}{\tau}}\sum_{t=1}^{T} 1/t + ...\right] \\
&\leq \alpha\left[\sum_{\theta=0}^{T} e^{-\frac{\theta}{\tau}}(1+\ln(T))\right] \quad \text{since} \sum_{t=1}^{T}\frac{1}{T} \leq 1+\ln(T) \\
&\leq \alpha(1+\ln(T))(1+\int_{0}^{\infty} e^{-\frac{\theta}{\tau}}d\theta) \\
&= \alpha(1+\ln(T))(1+\tau).
\end{aligned}$$

□

We are now ready to prove Theorem 27.

Proof of Theorem 27. Combining Eq (5.3), Lemma 46, Lemma 48 and Lemma 53, we have

$$\begin{aligned}
&\sup_{\pi\in\Pi} R(T,\pi) \\
&\leq (2\tau+2)+\left[\max_{\pi\in\Delta_M}\sum_{t=1}^{T}\langle\mu^{\pi}, r_t\rangle - \sum_{t=1}^{T}\langle\mu^{\pi_t}, r_t\rangle\right] \\
&\quad +\left[\sum_{t=1}^{T} 2e^{-\frac{t-1}{\tau}} + \sum_{t=1}^{T}\sum_{\theta=0}^{t-1} e^{-\frac{\theta}{\tau}}\|\mu^{\pi_{t-\theta}} - \mu^{\pi_{t-(\theta+1)}}\|_1\right] \\
&\leq 4(\tau+1)+\left[\max_{\pi\in\Delta_M}\sum_{t=1}^{T}\langle\mu^{\pi}, r_t\rangle - \sum_{t=1}^{T}\langle\mu^{\pi_t}, r_t\rangle\right] + 2\eta\left(1+\frac{1}{\eta}G_R\right)(1+\ln(T))(1+\tau). \quad (5.5)
\end{aligned}$$

The second term in Eq (5.5) is bounded by

$$
\begin{aligned}
&\max_{\pi\in\Delta_M}\sum_{t=1}^{T}\langle\mu^{\pi},r_t\rangle-\sum_{t=1}^{T}\langle\mu^{\pi_t},r_t\rangle\\
&\leq \max_{\pi\in\Delta_{M,\delta}}\sum_{t=1}^{T}\langle\mu^{\pi},r_t\rangle-\sum_{t=1}^{T}\langle\mu^{\pi_t},r_t\rangle+2\delta T\left(|S||A|-1\right) \quad \text{Lemma 52}\\
&\leq \sum_{t=1}^{T}\langle\mu^{\pi_{t+1}},r_t\rangle-\sum_{t=1}^{T}\langle\mu^{\pi_t},r_t\rangle+\frac{T}{\eta}\max_{\mu_1,\mu_2\in\Delta_{M,\delta}}\left[R(\mu_1)-R(\mu_2)\right]+2\delta T\left(|S||A|-1\right)\\
&\qquad \text{by Lemma 49}\\
&\leq \sum_{t=1}^{T}\langle\mu^{\pi_{t+1}},r_t\rangle-\sum_{t=1}^{T}\langle\mu^{\pi_t},r_t\rangle+\frac{T}{\eta}\ln(|S||A|)+2\delta T\left(|S||A|-1\right)\\
&\qquad \text{by choice of function } R\\
&\leq \sum_{t=1}^{T}\|r_t\|_\infty\|\mu^{\pi_{t+1}}-\mu^{\pi_t}\|_1+\frac{T}{\eta}\ln(|S||A|)+2\delta T\left(|S||A|-1\right)\\
&\qquad \text{by Cauchy-Schwarz inequality}\\
&\leq \sum_{t=1}^{T}\frac{2\eta}{t}\left(1+\frac{1}{\eta}G_R\right)+\frac{T}{\eta}\ln(|S||A|)+2\delta T\left(|S||A|-1\right) \quad \text{by Lemma 50}\\
&\leq 2\eta\left(1+\frac{1}{\eta}G_R\right)\left(1+\ln(T)\right)+\frac{T}{\eta}\ln(|S||A|)+2\delta T\left(|S||A|-1\right).
\end{aligned}
$$

Plugging this result into Eq (5.5), we get

$$
\begin{aligned}
\sup_{\pi\in\Pi}R(T,\pi) &\leq 4(\tau+1)+2\eta(1+\frac{1}{\eta}G_R)(1+\ln(T))+\frac{T}{\eta}\ln(|S||A|)\\
&\qquad +2\delta T\left(|S||A|-1\right)+2\eta(1+\frac{1}{\eta}G_R)(1+\ln(T))(1+\tau)\\
&\leq 4(\tau+1)+4\eta(1+\frac{1}{\eta}G_R)(1+\ln(T))(1+\tau)\\
&\qquad +\frac{T}{\eta}\ln(|S||A|)+2\delta T\left(|S||A|-1\right)\\
&= O\left(\tau+4\sqrt{\tau T\ln(|S||A|)}\ln(T)+\sqrt{\tau T\ln(|S||A|)}+e^{-\frac{\sqrt{T}}{\sqrt{\tau}}}T|S||A|\right).
\end{aligned}
$$

The proof is completed by choosing $\eta=\sqrt{\frac{T\ln(|S||A|)}{\tau}}$ and $\delta=e^{-\frac{\sqrt{T}}{\sqrt{\tau}}}$, and using the fact

that $G_R \leq \max\{|\ln(\delta)|, 1\}$. $\square$

5.6 Online MDPs with Large State Space

In the previous section, we designed an algorithm for Online MDP with sublinear regret.

However, the computational complexity of our algorithm is $O(poly(|S||A|))$ per round. MDPs in practice often have extremely large state space S due to the curse of dimenionality [26], so computing the exact solution becomes impractical. In this section, we propose an approximate algorithm that can handle large state space.

5.6.1 Approximating Occupancy Measures and Regret Definition

We consider an approximation scheme introduced in [2] for standard MDPs. The idea is to use d feature vectors (with $d \ll |S||A|$) to approximate occupancy measures $\mu \in \mathbb{R}^{|S|\times|A|}$. Specifically, we approximate $\mu \approx \Phi\theta$ where Φ is a given matrix of dimension $|S||A| \times d$, and $\theta \in \Theta \triangleq \{\theta \in \mathbb{R}^d_+ : \|\theta\|_\infty \leq W\}$ for some positive constant W.

As we will restrict the occupancy measures chosen by our algorithm to satisfy $\mu = \Phi\theta$, the definition of MDP-regret (5.1) is too strong as it compares against all stationary policies. Instead, we restrict the benchmark to be the set of policies Π^Φ that can be represented by matrix Φ, where

$$\Pi^\Phi \triangleq \{\pi \in \Pi : \text{ there exists } \mu^\pi \in \Delta_M \text{ such that } \mu^\pi = \Phi\theta \text{ for some } \theta \in \Theta\}.$$

Our goal will now be to achieve sublinear Φ-MDP-regret defined as

$$\Phi\text{-MDP-Regret}(T) \triangleq \max_{\pi\in\Pi^\Phi} \mathbb{E}[\sum_{t=1}^{T} r_t(s_t^\pi, a_t^\pi)] - \mathbb{E}[\sum_{t=1}^{T} r_t(s_t, a_t)], \tag{5.6}$$

where the expectation is taken with respect to random state transitions of the MDP and randomization used in the algorithm. Additionally, we want to make the computational complexity *independent* of $|S|$ and $|A|$.

Choice of Matrix Φ and Computation Efficiency. The columns of matrix $\Phi \in \mathbb{R}^{|S||A|\times d}$ represent probability distributions over state-action pairs. The choice of Φ is problem-dependent, and a detailed discussion is beyond the scope of this chapter. Abbasi-Yadkori et al. [2] shows that for many applications such as the game of Tetris and queuing networks, Φ can be naturally chosen as a sparse matrix, which allows constant time access to entries of Φ and efficient dot product operations. We will assume such constant time access throughout our analysis.

We refer readers to [2] for further details.

5.6.2 The Approximate Algorithm

The algorithm we propose is built on MDP-RFTL, but is significantly modified in several aspects. We start with key ideas on how and why we need to modify the previous algorithm, and then formally present the new algorithm.

To aid our analysis, we make the following definition.

Definition 7. *Let $\tilde{\delta}_0 \geq 0$ be the largest real number such that for all $\delta \in [0, \tilde{\delta}_0]$ the set $\Delta^{\Phi}_{M,\delta} \triangleq \{\mu \in \mathbb{R}^{|S||A|} : \text{ there exists } \theta \in \Theta \text{ such that } \mu = \Phi\theta, \mu \geq \delta, \mu^\top 1 = 1, \mu^\top(P - B) = 0\}$ is nonempty. We also write $\Delta^{\Phi}_{M} \triangleq \Delta^{\Phi}_{M,0}$.*

As a first attempt, one could replace the shrunk set of occupancy measures $\Delta_{M,\delta}$ in Algorithm 13 with $\Delta^{\Phi}_{M,\delta}$ defined above. We then use occupancy measures $\mu^{\Phi\theta^*_{t+1}} \triangleq \Phi\theta^*_{t+1}$ given by the RFTL algorithm, i.e.,

$\theta^*_{t+1} \leftarrow \arg\max_{\theta\in\Delta^{\Phi}_{M,\delta}} \sum_{i=1}^{t} [\langle r_i, \mu\rangle - (1/\eta)R(\mu)]$. The same proof of Theorem 27 would apply and guarantee a sublinear Φ-MDP-Regret.

Unfortunately, replacing $\Delta_{M,\delta}$ with $\Delta^{\Phi}_{M,\delta}$ does not reduce the time complexity of computing the iterates $\{\mu^{\Phi\theta^*_t}\}_{t=1}^{T}$, which is still $poly(|S||A|)$.

To tackle this challenge, we will not apply the RFTL algorithm exactly, but will instead obtain an approximate solution in $poly(d)$ time. We relax the constraints $\mu \geq \delta$ and $\mu^\top(P-$

$B) = 0$ that define the set $\Delta^{\Phi}_{M,\delta}$, and add the following penalty term to the objective function:

$$V(\theta) \triangleq -H_t \|(\Phi\theta)^\top (P - B)\|_1 - H_t \|\min\{\delta, \Phi\theta\}\|_1. \tag{5.7}$$

Here, $\{H_t\}_{t=1}^T$ is a sequence of tuning parameters that will be specified in Theorem 28. Let $\Theta^\Phi \triangleq \{\theta \in \Theta\,, \mathbf{1}^\top(\Phi\theta) = 1\}$. Thus, the original RFTL step in Algorithm 13 now becomes

$$\max_{\theta \in \Theta^\Phi} \sum_{i=1}^{t} c^{t,\eta}(\theta), \quad \text{where } c^{t,\eta}(\theta) \triangleq \sum_{i=1}^{t} \left[\langle r_i, \Phi\theta\rangle - \frac{1}{\eta} R^\delta(\Phi\theta)\right] + V(\theta). \tag{5.8}$$

In the above function, we use a modified entropy function $R^\delta(\cdot)$ as the regularization term, because the standard entropy function has an infinite gradient at the origin. More specifically, let $R_{(s,a)}(\mu) \triangleq \mu(s,a)\ln(\mu(s,a))$ be the entropy function. We define $R^\delta(\mu) = \sum_{(s,a)} R^\delta_{(s,a)}(\mu(s,a))$, where

$$R^\delta_{(s,a)} \triangleq \begin{cases} R_{(s,a)}(\mu) & \text{if } \mu(s,a) \geq \delta \\ R_{(s,a)}(\delta) + \frac{d}{d\mu(s,a)} R_{(s,a)}(\delta)(\mu(s,a) - \delta) & \text{otherwise.} \end{cases} \tag{5.9}$$

Since computing an exact gradient for function $c^{t,\eta}(\cdot)$ would take $O(|S||A|)$ time, we solve problem (5.8) by stochastic gradient ascent. The following lemma shows how to efficiently generate stochastic subgradients for function $c^{t,\eta}$ via sampling.

Lemma 54. *Let q_1 be any probability distribution over state-action pairs, and q_2 be any probability distribution over all states. Sample a pair $(s', a') \sim q_1$ and $s'' \sim q_2$. The quantity*

$$g_{s',a',s''}(\theta) = \Phi^\top r_t + \frac{H_t}{q_1(s',a')} \Phi_{(s',a'),:} \mathbb{I}\{\Phi_{(s',a'),:}\theta \leq \delta\} \\ - \frac{H_t}{q_2(s'')} [(P - B)^\top \Phi]_{s'',:} sign([(P - B)^\top \Phi]_{s'',:}\theta) - \frac{t}{\eta q_1(s',a')} \nabla_\theta R^\delta_{(s',a')}(\Phi\theta)$$

satisfies $\mathbb{E}_{(s',a')\sim q_1, s''\sim q_2}[g_{s',a',s''}(\theta)|\theta] = \nabla_\theta c^{\eta,t}(\theta)$ for any $\theta \in \Theta$. Morever, we have

$\|g(\theta)\|_2 \leq t\sqrt{d} + H_t(C_1 + C_2) + \frac{t}{\eta}(1 + \ln(Wd) + |\ln(\delta)|)C_1$, *w.p.1, where*

$$C_1 = \max_{(s,a)\in S\times A} \frac{\|\Phi_{(s,a),:}\|_2}{q_1(s,a)}, \quad C_2 = \max_{s\in S} \frac{\|(P-B)^\top_{:,s}\Phi\|_2}{q_2(s)}. \tag{5.10}$$

Putting everything together, we present the complete approximate algorithm for large state online MDPs in Algorithm 14. The algorithm uses Projected Stochastic Gradient Ascent (Algorithm 15) as a subroutine, which uses the sampling method in Lemma 54 to generate stochastic sub-gradients.

Algorithm 14 (LARGE-MDP-RFTL)

input: matrix Φ, parameters: $\eta, \delta > 0$, convex function $R^\delta(\mu)$, SGA step-size schedule $\{w_t\}_{t=0}^T$, penalty term parameters $\{H_t\}_{t=1}^T$
initialize: $\tilde{\theta}_1 \leftarrow \text{PSGA}(-R^\delta(\Phi\theta) + V(\theta), \Theta^\Phi, w_0, K_0)$
for $t = 1, ..., T$ **do**
  observe current state s_t; play action a with distribution $\frac{[\Phi\tilde{\theta}_t]_+(s_t,a)}{\sum_{a\in A}[\Phi\tilde{\theta}_t]_+(s_t,a)}$
  observe $r_t \in [-1,1]^{|S||A|}$
  $\tilde{\theta}_{t+1} \leftarrow \text{PSGA}(\sum_{i=1}^t [\langle r_i, \Phi\theta\rangle - \frac{1}{\eta}R^\delta(\Phi\theta)] + V(\theta), \Theta^\Phi, w_t, K_t)$
end for

Algorithm 15 Projected Stochastic Gradient Ascent: $\text{PSGA}(f, X, w, K)$

input: concave objective function f, feasible set X, stepsize w, $x_1 \in X$
for $k = 1, ...K$ **do**
  compute a stochastic subgradient g_k such that $\mathbb{E}[g_k] = \nabla f(x_k)$ using Lemma 54
  set $x_{k+1} \leftarrow P_X(x_k + wg(x_k))$
end for
output: $\frac{1}{K}\sum_{k=1}^K x_k$

5.6.3 Sketch of Analysis of the Approximate Algorithm

We establish a regret bound for the LARGE-MDP-RFTL algorithm as follows.

Theorem 28. *Suppose* $\{r_t\}_{t=1}^T$ *is an arbitrary sequence of rewards such that* $|r_t(s,a)| \leq 1$ *for all* $s \in S$ *and* $a \in A$. *For* $T \geq \ln^2(\frac{1}{\delta_0})$, LARGE-MDP-RFTL *with parameters* $\eta = \sqrt{\frac{T}{\tau}}, \delta = e^{-\sqrt{T}}$, $K(t) = \left[W^{3/2}t^2d^{3/2}\tau^4(C_1+C_2)T^{3/2}\ln(WTd)\right]^2$, $w_t =$

$\frac{\sqrt{d}W}{\sqrt{K(t)}(t\sqrt{d}+H_t(C_1+C_2)+\frac{t}{\eta}C_1)}$ *guarantees that*

$$\Phi\text{-}\textit{MDP-Regret}(T) \leq O(c_{S,A}\ln(|S||A|)\sqrt{\tau T}\ln(T)).$$

Here $c_{S,A}$ is a problem dependent constant. The constants C_1, C_2 are defined in Lemma 54.

A salient feature of the LARGE-MDP-RFTL algorithm is that its computational complexity in each period is independent of the size of state space $|S|$ or the size of action space $|A|$, and thus is amenable to large scale MDPs. In particular, in Theorem 28, the number of SGA iterations, $K(t)$, is $O(d)$ and independent of $|S|$ and $|A|$.

Compared to Theorem 27, we achieve a regret with similar dependence on the number of periods T and the mixing time τ. The regret bound also depends on $\ln(|S|)$ and $\ln(|A|)$, with an additional constant term $c_{S,A}$. The constant comes from a projection problem (see details in Section 5.7) and may grow with $|S|$ and $|A|$ in general. But for some MDP problems, $c_{S,A}$ can be bounded by an absolute constant: an example is the well-known (Markovian) multi-armed bandit problem [133].

For a more detailed discussion of the constant $c_{S,A}$, we refer readers to Appendix C.1.

Proof Idea for Theorem 28. Consider the MDP-RFTL iterates ,$\{\theta_t^*\}_{t=1}^T$, and the occupancy measures $\{\mu^{\Phi\theta_t^*}\}_{t=1}^T$ induced by following policies $\{\Phi\theta_t^*\}_{t=1}^T$. Since $\theta_t^* \in \Delta_{M,\delta}^\Phi$ it holds that $\mu^{\Phi\theta_t^*} = \Phi\theta_t^*$ for all t. Thus, following the proof of Theorem 27, we can obtain the same Φ-MDP-Regret bound in Theorem 27 if we follow policies $\{\Phi\theta_t^*\}_{t=1}^T$. However, computing θ_t^* takes $O(poly(|S||A|))$ time.

The crux the proof of Theorem 28 is to show that the $\{\Phi\tilde{\theta}_t\}_{t=1}^T$ iterates in Algorithm 14 induce occupancy measures $\{\mu^{\Phi\tilde{\theta}_t}\}_{t=1}^T$ that are close to $\{\mu^{\Phi\theta_t^*}\}_{t=1}^T$. Since the algorithm has relaxed constraints of $\Delta_{M,\delta}^\Phi$, in general we have $\Phi\tilde{\theta}_t \notin \Delta_{M,\delta}^\Phi$ and thus $\mu^{\Phi\tilde{\theta}_t} \neq \Phi\tilde{\theta}_t$. So we need to show that the distance between $\mu^{\Phi\theta_{t+1}^*}$, and $\mu^{\Phi\tilde{\theta}_{t+1}}$ is small. Using triangle

inequality we have

$$\|\mu^{\Phi\theta_t^*} - \mu^{\Phi\tilde{\theta}_t}\|_1 \leq \|\mu^{\Phi\theta_t^*} - P_{\Delta^{\Phi}_{M,\delta}}(\Phi\tilde{\theta}_t)\|_1 + \|P_{\Delta^{\Phi}_{M,\delta}}(\Phi\tilde{\theta}_t) - \Phi\tilde{\theta}_t\|_1 + \|\Phi\tilde{\theta}_t - \mu^{\Phi\tilde{\theta}_t}\|_1,$$

where $P_{\Delta^{\Phi}_{M,\delta}}(\cdot)$ denotes the Euclidean projection onto $\Delta^{\Phi}_{M,\delta}$. We then proceed to bound each term individually. We defer the details to Section 5.7 as bounding each term requires lengthy proofs.

5.7 Regret Analysis of the Approximate Algorithm

Using Lemma 46 and Lemma 48 in Section 5.5, we can obtain a bound on Φ-MDP-Regret as follows.

$$\begin{aligned}
&\max_{\pi\in\Pi^{\Phi}} R(\pi, T) \leq \mathbb{E}\left[(2\tau+2)+\max_{\pi\in\Pi^{\Phi}}\left[\sum_{t=1}^{T}\rho_t^{\pi}-\sum_{t=1}^{T}\rho_t\right]+\left[\sum_{t=1}^{T}\rho_t-\mathbb{E}[\sum_{t=1}^{T}r_t(s_t,a_t)]\right]\right]\\
=&\mathbb{E}\left[(2\tau+2)+\left[\max_{\mu\in\Delta^{\Phi}_{M,\delta}}\sum_{t=1}^{T}\langle\mu, r_t\rangle-\sum_{t=1}^{T}\langle\mu^{\Phi\tilde{\theta}_t}, r_t\rangle\right]+\left[\sum_{t=1}^{T}\rho_t-\mathbb{E}[\sum_{t=1}^{T}r_t(s_t,a_t)]\right]\right]\\
\leq&\mathbb{E}\left[2(2\tau+2)+\left[\max_{\mu\in\Delta^{\Phi}_{M,\delta}}\sum_{t=1}^{T}\langle\mu, r_t\rangle-\sum_{t=1}^{T}\langle\mu^{\Phi\tilde{\theta}_t}, r_t\rangle\right]+\left[\sum_{t=1}^{T}\sum_{i=0}^{t-i}e^{-\frac{i}{\tau}}\|\mu^{\Phi\tilde{\theta}_{t-i}}-\mu^{\Phi\tilde{\theta}_{t-(i+1)}}\|_1\right]\right].
\end{aligned}$$

Let θ_t^* be a solution to the following optimization problem:

$$\begin{aligned}
&\max_{\theta\in\Theta}\ \sum_{i=1}^{t-1}\left[\langle\mu, r_i\rangle+\frac{1}{\eta}R^{\delta}(\mu)\right]\\
&\text{s.t}\quad \mu = \Phi\theta\\
&\qquad\sum_{s\in S}\sum_{a\in A}\mu(s,a)P(s'|s,a) = \sum_{a\in A}\mu(s',a)\quad \forall s'\in S\\
&\qquad\sum_{s\in S}\sum_{a\in A}\mu(s,a) = 1\\
&\qquad\mu(s,a)\geq 0\quad \forall s\in S, \forall a\in A.
\end{aligned}$$

Since $\{\Phi\theta_t^*\}_{t=1}^{T}$ represents the iterates of RFTL, we can use the regret guarantee of

RFTL to bound $\max_{\mu \in \Delta^{\Phi}_{M,\delta}} \sum_{t=1}^{T} \langle \mu, r_t \rangle - \sum_{t=1}^{T} \langle \mu^{\Phi\theta^*_t}, r_t \rangle$. Notice also that $\mu^{\Phi\theta^*_t} = \Phi\theta^*_t$ as θ^*_t satisfies all the constraints that ensure $\Phi\theta^*_t$ is an occupancy measure.

In the remainder of the proof, we want to show that the occupancy measures $\mu^{\Phi\tilde{\theta}_t}$ induced by our algorithm's iterates $\Phi\tilde{\theta}_t$ are close to $\mu^{\Phi\theta^*_t}$.

The rest of the analysis is to prove that $\|\mu^{\Phi\theta^*_t} - \mu^{\Phi\tilde{\theta}_t}\|_1$ is small. Notice that using the triangle inequality, we can upper bound this distance by

$$\begin{aligned}\|\mu^{\Phi\theta^*_t} - \mu^{\Phi\tilde{\theta}_t}\|_1 &\leq \|\mu^{\Phi\theta^*_t} - P_{\Delta^{\Phi}_{M,\delta}}(\Phi\tilde{\theta}_t)\|_1 + \|P_{\Delta^{\Phi}_{M,\delta}}(\Phi\tilde{\theta}_t) - \Phi\tilde{\theta}_t\|_1 + \|\Phi\tilde{\theta}_t - \mu^{\Phi\tilde{\theta}_t}\|_1 \\ &= \|\Phi\theta^*_t - P_{\Delta^{\Phi}_{M,\delta}}(\Phi\tilde{\theta}_t)\|_1 + \|P_{\Delta^{\Phi}_{M,\delta}}(\Phi\tilde{\theta}_t) - \Phi\tilde{\theta}_t\|_1 + \|\Phi\tilde{\theta}_t - \mu^{\Phi\tilde{\theta}_t}\|_1.\end{aligned}$$

To bound the last term, the following lemma from [2] will be useful. It relates a vector $\Phi\tilde{\theta}$ which is almost feasible with its occupancy measure.

Lemma 55. *[Lemma 2 in [2]] Let $u \in \mathbb{R}^{|S||A|}$ be a vector. Let $\mathcal{N}$ be the set of entries (s,a) where $u(s,a) \leq 0$. Assume*

$$\sum_{(s,a)} u(s,a) = 1, \qquad \sum_{(s,a)\in\mathcal{N}} |u(s,a)| \leq \epsilon', \quad \|u^\top (P - B)\|_1 \leq \epsilon''.$$

Vector $[u]_+ / \|[u]_+\|_1$ defines a policy, which in turn defines a stationary distribution μ^u. It holds that

$$\|\mu^u - u\|_1 \leq \tau \ln(\frac{1}{\epsilon'})(2\epsilon' + \epsilon'') + 3\epsilon'.$$

Suppose we are given a vector $\Phi\tilde{\theta}_t$ such that $\|[\Phi\tilde{\theta}_t]_{(\delta,-)}\|_1 \leq \epsilon'$ and $\|(\Phi\tilde{\theta}_t)^\top (P - B)\|_1 \leq \epsilon''$. In view of Lemma 55 and the fact that $\|[\Phi\tilde{\theta}_t]_-\|_1 \leq \|[\Phi\tilde{\theta}_t]_{(\delta,-)}\|_1 \leq \epsilon'$, we have a bound on $\|\Phi\tilde{\theta}_t - \mu^{\Phi\tilde{\theta}_t}\|_1$. The next lemma shows that we can also obtain a bound on $\|P_{\Delta^{\Phi}_{M,\delta}}(\Phi\tilde{\theta}_t) - \Phi\tilde{\theta}_t\|_1$.

Lemma 56. *Let $\Phi\tilde{\theta}_t$ be a vector such that $\|[\Phi\tilde{\theta}]_{(\delta,-)}\|_1 \leq \epsilon'$ and $\|(\Phi\tilde{\theta})^\top (P - B)\|_1 \leq \epsilon''$*

for some $\epsilon', \epsilon'' \geq 0$. It holds that

$$\|P_{\Delta^{\Phi}_{M,\delta}}(\Phi\tilde{\theta}_t) - \Phi\tilde{\theta}_t\|_1 \leq c(\epsilon' + \epsilon''),$$

where c is a bound on the l_∞ norm of the Lagrange multipliers of certain linear programming problem.

Proof. The idea comes from sensitivity analysis in Linear Programming (LP) (see for example [115]). Consider the l_1 projection problem of $\Phi\tilde{\theta}_t$ onto the set of occupancy measures parametrized by Φ

$$\begin{aligned}
&\min_{\theta} \|\mu - \Phi\tilde{\theta}\|_1 \\
\text{s.t} \quad &\mu = \Phi\theta \\
&\mu^\top 1 = 1 \\
&\mu \geq \delta \\
&\mu^\top (P - B) = 0 \\
&\theta \in \Theta.
\end{aligned}$$

It can be reforumulated as the following LP

$$\begin{aligned}
\text{Primal 1:} \qquad &\min_{\theta, u} \sum_{(s,a)} u(s,a) \\
\text{s.t} \quad &u(s,a) - [\Phi\theta](s,a) \geq -[\Phi\tilde{\theta}](s,a) \\
&u(s,a) + [\Phi\theta](s,a) \geq [\Phi\tilde{\theta}](s,a) \\
&\mu = \Phi\theta \\
&\mu^\top 1 = 1 \\
&\mu \geq \delta \\
&\mu^\top (P - B) = 0
\end{aligned}$$

$$-\theta(i) \geq -W \quad \forall i = 1, ..., d$$

$$\theta(i) \geq 0 \quad \forall i = 1, ..., d$$

Let us now consider the perturbed problem 'Primal 2' which arises by perturbing the right hand side vector of 'Primal 1':

$$\begin{aligned}
\text{Primal 2:} \qquad & \min_{\theta, u} \sum_{(s,a)} u(s,a) \\
\text{s.t} \quad & u(s,a) - [\Phi\theta](s,a) \geq -[\Phi\tilde{\theta}](s,a) \\
& u(s,a) + [\Phi\theta](s,a) \geq [\Phi\tilde{\theta}](s,a) \\
& \mu = \Phi\theta \\
& \mu^\top 1 = 1 \\
& \mu \geq \delta + \tilde{a} \\
& \mu^\top (P - B) = \tilde{b} \\
& -\theta(i) \geq -W \quad \forall i = 1, ..., d \\
& \theta(i) \geq 0 \quad \forall i = 1, ..., d
\end{aligned}$$

We choose perturbation vectors $\tilde{a}, \tilde{b}$ such that the optimal value of ' Primal 2' is 0. Let b be the right hand side vector of 'Primal 1' and $b' \triangleq b - \xi$ be that of 'Primal 2' for some vector ξ. Since by assumption we have that $\|[\Phi\tilde{\theta}]_{(\delta,-)}\|_1 \leq \epsilon'$ and $\|(\Phi\tilde{\theta})^\top (P - B)\|_1 \leq \epsilon''$ then it holds that $\|b - b'\|_1 = \|\xi\|_1 \leq \epsilon' + \epsilon''$. Let 'Opt. Primal 1' and 'Opt. Primal 2' be the optimal value of the respective problems ('Opt. Primal 2' = 0 by construction) and let λ^* be the vector of optimal dual variables of 'Dual 1', the problem dual to 'Primal 1'. Since by assumption, the feasible set of 'Primal 1' is feasible, then the absolute value of the entries of λ^* is bounded by some constant c.

Now, since λ^* is feasible for 'Dual 2', the following sequence of inequalities hold:

$$\begin{aligned} &\text{'Opt. Primal 2'} \geq (\lambda^*)^\top (b - \xi) \\ \iff &\text{'Opt. Primal 2'} \geq \text{'Opt. Primal 1'} - (\lambda^*)^\top \xi. \end{aligned}$$

Therefore,

$$\begin{aligned} \text{'Opt. Primal 1'} &\leq \text{'Opt. Primal 2'} + \|\lambda^*\|_\infty \|\xi\|_\infty \\ &= 0 + \|\lambda^*\|_\infty \|\xi\|_1 \\ &\leq c(\epsilon' + \epsilon''), \end{aligned}$$

which yields the result. □

Now, we proceed to bound $\|\Phi\theta_t^* - P_{\Delta_{M,\delta}^\Phi}(\Phi\tilde{\theta}_t)\|_1$. Consider the function

$$F_t(\Phi\theta) \triangleq \sum_{i=1}^{t} [\langle r_i, \Phi\theta \rangle - \frac{1}{\eta} R^\delta(\Phi\theta)]. \tag{5.11}$$

Since R^δ is strongly convex over $\Delta_{M,\delta}^\Phi$ with respect to $\|\cdot\|_1$ (but not everywhere over the reals as the extension uses a linear function), we have that F_t is $\frac{t}{\eta}$-strongly concave with respect to $\|\cdot\|_1$ over $\Delta_{M,\delta}^\Phi$. With this in mind, we can prove the following result.

Lemma 57. *Let $\Phi\tilde{\theta}_{t+1}$ be a vector such that $\|[\Phi\tilde{\theta}_{t+1}]_{(\delta,-)}\|_1 \leq \epsilon'$ and $\|(\Phi\tilde{\theta}_{t+1})^\top (P - B)\|_1 \leq \epsilon''$ for some $\epsilon', \epsilon'' \geq 0$. Let ϵ''' be such that $F_t(\Phi\theta_{t+1}^*) - F_t(\Phi\tilde{\theta}_{t+1}) \leq \epsilon'''$. And let G_{F_t} be the Lipschitz constant of F_t with respect to norm $\|\cdot\|_1$ over the set $\Delta_{M,\delta}^\Phi$. It holds that*

$$\|\Phi\theta_{t+1}^* - P_{\Delta_{M,\delta}^\phi}(\Phi\tilde{\theta}_{t+1})\|_1 \leq \sqrt{\frac{2\eta}{t}(\epsilon''' + G_{F_t} c(\epsilon' + \epsilon''))}.$$

Proof. Since F_t is $\frac{t}{\eta}$-strongly concave over $\Delta^{\Phi}_{M,\delta}$ and $\Phi\theta^*_{t+1}$ is the optimizer of F_t over $\Delta^{\Phi}_{M,\delta}$. It holds that

$$\begin{aligned}
\frac{t}{2\eta}\|\Phi\theta^*_{t+1} - \Phi\tilde{\theta}_{t+1}\|_1^2 &\leq F_t(\Phi\theta^*_{t+1}) - F_t(P_{\Delta^{\Phi}_{M,\delta}}(\Phi\tilde{\theta}_{t+1})) \\
&\leq F_t(\Phi\theta^*_{t+1}) - F_t(\Phi\tilde{\theta}_{t+1}) + G_{F_t}\|P_{\Delta^{\Phi}_{M,\delta}}(\Phi\tilde{\theta}_{t+1}) - \Phi\tilde{\theta}_{t+1}\|_1 \\
&\leq \epsilon''' + G_{F_t}\|P_{\Delta^{\Phi}_{M,\delta}}(\Phi\tilde{\theta}_{t+1}) - \Phi\tilde{\theta}_{t+1}\|_1 \quad \text{by assumption} \\
&\leq \epsilon''' + G_{F_t}c(\epsilon' + \epsilon'') \quad \text{by Lemma 56}
\end{aligned}$$

which yields the result. □

The next lemma bounds the Lipschitz constant G_{F_t}.

Lemma 58. *Let* $\eta = \sqrt{\frac{T}{\tau}}$, $\delta = e^{-\sqrt{T}}$. *The function* $F_t(\mu) : \mathbb{R}^{|S||A|} \to \mathbb{R}$ *is* G_{F_t}*-Lipschitz continuous on variables* μ *with respect to norm* $\|\cdot\|_1$ *over* $\Delta^{\Phi}_{M,\delta}$ *with* $G_{F_t} \leq t(1 + 2\sqrt{\tau}\ln(dW))$.

Proof. It suffices to find a an upper bound for $\|\nabla_\mu F_t(\mu)\|_\infty$. Since $\nabla_\mu F_t(\mu) = \sum_{i=1}^t r_i - \frac{t}{\eta}\nabla_\mu R^\delta(\mu)$, we have that

$$\begin{aligned}
\|\nabla_\mu F_t(\mu)\|_\infty &\leq \|\sum_{i=1}^t r_i\|_\infty + \frac{t}{\eta}\|\nabla_\mu R^\delta(\mu)\|_\infty \quad \text{by triangle inequality} \\
&\leq t + \frac{t}{\eta}\|\nabla_\mu R^\delta(\mu)\|_\infty \quad \text{since } |r_i(s,a)| \leq 1 \\
&\leq t + \frac{t}{\eta}\max\{|1 + \ln(\delta)|, |1 + \ln(dW)|\} \quad \text{as in the proof of Lemma 51 .}
\end{aligned}$$

The second to last inequality holds since $|\frac{d}{dx}x\ln(x)| = |1 + \ln(x)|$ and the maximum will occur at $x = \delta$ or $x = [\Phi\theta](s,a)$, $[\Phi\theta](s,a)$ can be bounded by Wd. Plugging in the values for η and δ we get

$$\|\nabla_\mu F_t(\mu)\|_\infty \leq t + \frac{t\tau}{\sqrt{T}}(1 + \max\{\sqrt{T}, \ln(dW)\})$$

$$\leq t + \frac{t\tau}{\sqrt{T}}(2\sqrt{T}\ln(dW))$$
$$= t(1 + 2\sqrt{\tau}\ln(dW)).$$

□

Combining the previous three lemmas, we obtain the following result.

Lemma 59. *Let $\Phi\tilde{\theta}_{t+1}$ be a vector such that $\|[\Phi\tilde{\theta}_{t+1}]_{(\delta,-)}\|_1 \leq \epsilon'$ and $\|(\Phi\tilde{\theta}_{t+1})^\top(P - B)\|_1 \leq \epsilon''$ for some $\epsilon', \epsilon'' \geq 0$. Let ϵ''' be such that $F_t(\Phi\theta^*_{t+1}) - F_t(\Phi\tilde{\theta}_{t+1}) \leq \epsilon'''$. And let G_{F_t} be the Lipschitz constant of F_t with respect to norm $\|\cdot\|_1$ over the set $\Delta^\Phi_{M,\delta}$. It holds that*

$$\|\mu^{\Phi\theta^*_t} - \mu^{\Phi\tilde{\theta}_t}\|_1 \leq \tau\ln(\frac{1}{\epsilon'})(2\epsilon' + \epsilon'') + 3\epsilon' + c(\epsilon' + \epsilon'') + \sqrt{\frac{2\eta}{t}(\epsilon''' + G_{F_t}c(\epsilon' + \epsilon''))}.$$

Proof. By triangle inequality, we have

$$\|\mu^{\Phi\theta^*_t} - \mu^{\Phi\tilde{\theta}_t}\|_1 \leq \|\Phi\theta^*_t - P_{\Delta^\Phi_{M,\delta}}(\Phi\tilde{\theta}_t)\|_1 + \|P_{\Delta^\Phi_{M,\delta}}(\Phi\tilde{\theta}_t) - \Phi\tilde{\theta}_t\|_1 + \|\Phi\tilde{\theta}_t - \mu^{\Phi\tilde{\theta}_t}\|_1.$$

Using Lemmas 55, 56, and 57 to bound the first, second, and third terms respectively yields the result.

□

Now we can upper bound the bound on the Φ-MDP-Regret, Eq (5.11), using triangle inequality and Lemma 59. For the bound to be useful we want to be able to produce vectors $\{\Phi\tilde{\theta}_t\}_{t=1}^T$ that satisfy the conditions of Lemma 59 with $\epsilon', \epsilon'', \epsilon'''$ that are small enough. It is also important that we produce $\{\Phi\tilde{\theta}_t\}_{t=1}^T$ in a computationally efficient manner. At time t, our approach to generate $\Phi\tilde{\theta}_t$, will be to run Projected Stochastic Gradient Descent on function 5.8. The following theorem from [2] will be useful.

Theorem 29 (Theorem 3 in [2]). *Let $\mathcal{Z} \subset \mathbb{R}$ be a convex set such that $\|z\|_2 \leq Z$ for all $z \in \mathcal{Z}$ for some $Z > 0$. Let f be a concave function defined over $\mathcal{Z}$. Let $\{z_k\}_{k=1}^K \in \mathcal{Z}^T$ be the iterates of Projected Stochastic Gradient Ascent, i.e. $z_{k+1} \leftarrow P_{\mathcal{Z}}(x_k + \eta f'_t)$ where $P_{\mathcal{Z}}$ is the euclidean projection onto $\mathcal{Z}$, η is the step-size and $\{f'_k\}_{k=1}^K$ are such that $\mathbb{E}[f'_k|z_k] = \nabla f(z_k)$ with $\|f'_k\|_2 \leq F$ for some $F > 0$. Then, for $\eta = \frac{Z}{(F\sqrt{K})}$ for all $\kappa \in (0,1)$, with probability at least $1-\kappa$ it holds that*

$$\max_{z\in\mathcal{Z}} f(z) - f(\frac{1}{K}\sum_{k=1}^{K} z_k) \leq \frac{ZF}{\sqrt{K}} + \sqrt{\frac{(1+4Z^2K)\left(2\ln(\frac{1}{\kappa}) + d\ln(1+\frac{Z^2K}{d})\right)}{K^2}}.$$

In view of Theorem 29 we need to design a stochastic subgradient for $c^{t,\eta}$ and a bound for its l-2 norm. We follow the approach in [2], we notice however that the objective function considered in [2] does not contain the regularizer R^δ so must take care of that in our analysis.

Lemma 54 creates a stochastic subgradient for $c^{t,\eta}$ and provides an upper bound for its l-2 norm. We now present its proof.

Proof of Lemma 54. Let us first compute $\nabla_\theta c^{\eta,t}(\theta)$. Define $r_{:t} \triangleq \sum_{i=1}^t r_i$ By definition we have

$$\begin{aligned} c^{\eta,t}(\theta) &= (\Phi\theta)^\top r_{:t} - \frac{t}{\eta}\sum_{(s,a)} R^\delta_{(s,a)}(\Phi\theta) - H_t\|[\Phi\theta]_{(\delta,-)}\|_1 - H_t\|(P-B)^\top\Phi\theta\|_1 \\ &= \theta^\top(\Phi^\top r_{:t}) - \frac{t}{\eta}\sum_{(s,a)} R^\delta_{(s,a)}(\Phi\theta) \\ &\quad - H_t\sum_{(s,a)}[\Phi_{(s,a),:}\theta]_{(\delta,-)} - H_t\sum_s |[(P-B)^\top\Phi]_{s,:}\theta|. \end{aligned}$$

So, we get

$$\nabla_\theta c^{t,\eta}(\theta) = \Phi^\top r_{:t} - \frac{t}{\eta}\sum_{(s,a)} \nabla_\theta R^\delta_{(s,a)}(\Phi\theta) - H_t\sum_{(s,a)}$$

$$- \Phi_{(s,a),:}\mathbb{I}\{\Phi_{(s,a),:}\theta \leq \delta\} - H_t \sum_s [(P-B)^\top \Phi]_{s,:} sign([(P-B)^\top \Phi]_{s,:}\theta).$$

We design a stochastic subgradient g of $\nabla_\theta c^{\eta,t}(\theta)$ by sampling a state-action pair (s', a') from the given distribution q_1 and a state s'' from distribution q_2. Then, we have

$$\begin{aligned} g_{s',a',s''}(\theta) = \Phi^\top r_{:t} + \frac{H_t}{q_1(s',a')}\Phi_{(s',a'),:}\mathbb{I}\{\Phi_{(s',a'),:} \leq \delta\} \\ - \frac{H_t}{q_2(s'')}[(P-B)^\top \Phi]_{s'',:} sign([(P-B)^\top \Phi]_{s'',:}\theta) - \frac{t}{\eta q_1(s',a')}\nabla_\theta R^\delta_{(s',a')}(\Phi\theta). \end{aligned}$$

We will also give a closed form expression of $\nabla_\theta R^\delta_{(s',a')}(\Phi\theta)$ in the proof below. By construction, it holds that $\mathbb{E}_{(s',a')\sim q_1, s''\sim q_2}[g_{s',a',s''}(\theta)|\theta] = \nabla_\theta c^{t,\eta}(\theta)$. To simplify notation, let $g(\theta) = g_{s',a',s''}(\theta)$.

We now bound $\|g(\theta)\|_2$ with probability 1. First, we have

$$\begin{aligned} \|\Phi^\top r_{:t}\|_2 &= \sqrt{\sum_{i=1}^d (r_{:t}^\top \Phi_{:,i})^2} \\ &\leq \sqrt{\sum_{i=1}^d (\|r_{:t}\|_\infty \|\Phi_{:,i}\|_1)^2} \quad \text{by Cauchy-Schwarz} \\ &\leq \sqrt{dt^2 1} = t\sqrt{d}, \end{aligned}$$

where the last inequality holds since $\|r_i\|_\infty \leq 1$ for $t = 1, ..., T$ and each column of Φ is a probability distribution. Next, we have

$$\begin{aligned} &\left\| \frac{H_t}{q_1(s',a')}\Phi_{(s',a'),:}\mathbb{I}\{\Phi_{(s',a'),:}\theta \leq \delta\} \right\|_2 \leq H_t C_1, \quad \text{and} \\ &\left\| -\frac{H_t}{q_2(s'')}[(P-B)^\top \Phi]_{s'',:} sign([(P-B)^\top \Phi]_{s'',:}\theta) \right\|_2 \leq H_t C_2, \end{aligned}$$

where C_1 and C_2 are defined in (5.10). Finally, we bound $\|\nabla_\theta R^\delta_{(s,a)}(\Phi\theta)\|_2$. By definition of $R^\delta_{(s,a)}$ in Eq 5.9, we need to compute the gradients of the negative entropy function

$\nabla_\theta R(\Phi\theta)$. Let us compute $\frac{d}{d\theta_i} R(\Phi\theta)$ as follows.

$$\begin{aligned}
\frac{d}{d\theta_i} R(\Phi\theta) &= \sum_{(s,a)} \frac{d}{d\theta_i} R_{(s,a)}(\Phi\theta) \\
&= \sum_{(s,a)} \frac{d}{d\theta_i} \left[(\sum_{k=1}^{d} \Phi_{(s,a),k}\theta_k) \ln(\sum_{k=1}^{d} \Phi_{(s,a),k}\theta_k) \right] \\
&= \sum_{(s,a)} (\sum_{k=1}^{d} \Phi_{(s,a),k}\theta_k)(\frac{d}{d\theta_i} \ln(\sum_{k=1}^{d} \Phi_{(s,a),k}\theta_k)) + \ln(\sum_{k=1}^{d} \Phi_{(s,a),k}\theta_k)\Phi_{(s,a),i} \\
&= \sum_{(s,a)} (\sum_{k=1}^{d} \Phi_{(s,a)}\theta_k) \frac{1}{\sum_{k=1}^{d} \Phi_{(s,a)}\theta_k} \frac{d}{d\theta_i} (\sum_{k=1}^{d} \Phi_{(s,a),k}\theta_k) + \ln(\sum_{k=1}^{d} \Phi_{(s,a),k}\theta_k)\Phi_{(s,a),i} \\
&= \sum_{(s,a)} \Phi_{(s,a),i} + \ln(\sum_{k=1}^{d} \Phi_{(s,a),k}\theta_k)\Phi_{(s,a),i}.
\end{aligned}$$

We are also interested in the gradient of the linear extension of $R_{(s,a)}(x)$: $R_{(s,a)}(\delta) + \frac{d}{dx} R_{(s,a)}(\delta)(x - \delta)$ which is equal to $\delta \ln(\delta) + (1 + \ln(\delta))(x - \delta)$. So we upper bound $|\frac{d}{d\theta_i} \delta \ln(\delta) + (1 + \ln(\delta))(\Phi_{(s,a),:}\theta - \delta)|$ by

$$\begin{aligned}
&|\frac{d}{d\theta_i} \delta \ln(\delta) + (1 + \ln(\delta))(\Phi_{(s,a),:}\theta - \delta)| \\
=&|\frac{d}{d\theta_i} (1 + \ln(\delta))(\Phi_{(s,a),:}\theta - \delta)| \\
=&|(1 + \ln(\delta))\Phi_{(s,a),i}|.
\end{aligned}$$

It follows that

$$\begin{aligned}
&\|\nabla_\theta R^{\delta}_{(s,a)(\Phi\theta)}\|_2 \\
\leq& \left(\sum_{i=1}^{d} \left[\max\{\Phi_{(s,a),i} + \ln(W \sum_{k=1}^{d} \Phi_{(s,a),k})\Phi_{(s,a),i}, |(1 + \ln(\delta))\Phi_{(s,a),i}|\} \right]^2 \right)^{1/2} \\
\leq& \left(\sum_{i=1}^{d} \left[(1 + \max\{\ln(W \sum_{k=1}^{d} \Phi_{(s,a),k}), |\ln(\delta)|\}\Phi_{(s,a),i} \right]^2 \right)^{1/2}
\end{aligned}$$

$$\leq \left(\sum_{i=1}^{d} \left[(1 + \max\{\ln(Wd), |\ln(\delta)|\} \Phi_{(s,a),i} \right]^2 \right)^{1/2}$$

$$\leq (1 + \ln(Wd) + |\ln(\delta)|) \|\Phi_{(s,a),:}\|_2.$$

Thus, we have $\|\frac{t}{\eta q_1(s',a')} \nabla_\theta R^\delta_{(s',a')}(\Phi\theta)\|_2 \leq \frac{t}{\eta}(1 + \ln(Wd) + |\ln(\delta)|)C_1$. Using triangle inequality, we have that with probability 1

$$\|g(\theta)\|_2 \leq t\sqrt{d} + H(C_1 + C_2) + \frac{t}{\eta}(1 + \ln(Wd) + |\ln(\delta)|)C_1.$$

□

By using Lemma 54, as well as the fact that $\theta \in \Theta$ and $\|\theta\|_2 \leq d\|\theta\|_\infty$ implies $\|\theta\|_2 \leq W$, we can prove the following.

Lemma 60. *For all $t = 1, ..., T$, $\eta > 0$, $\kappa \in (0, 1)$, after running $K(t)$ iterations of Projected Stochastic Gradient Ascent on function $c^{\eta,t}(\theta)$ over the set Θ^Φ and using step-size $\frac{\sqrt{d}W}{\sqrt{K(t)G'}}$ with $G' = t\sqrt{d} + H_t(C_1 + C_2) + \frac{t}{\eta}(1 + \ln(Wd) + |\ln(\delta)|)C_1$ with probability at least $1 - \kappa$ it holds that*

$$\sum_{i=1}^{t} \left[\langle r_i, \Phi\theta^*_{t+1} \rangle - \frac{1}{\eta} R^\delta(\Phi\theta^*_{t+1}) \right]$$
$$- \left[\sum_{i=1}^{t} \left[\langle r_i, \Phi\tilde{\theta}_{t+1} \rangle - \frac{1}{\eta} R^\delta(\Phi\tilde{\theta}_{t+1}) \right] - H_t \|(\Phi\tilde{\theta}_{t+1})^\top (P - B)\|_1 - H_t \|[\Phi\tilde{\theta}_{t+1}]_{(\delta,-)}\|_1 \right]$$
$$\leq \frac{\sqrt{d}WG'}{\sqrt{K(t)}} + \sqrt{\frac{(1 + 4dW^2K(t))(2\ln(\frac{1}{\kappa}) + d\ln(1 + \frac{dW^2K(t)}{d}))}{K(t)^2}}.$$

Proof. The proof follows from applying Theorem 29 on function $c^{\eta,t}(\theta)$. Using the bound of the stochastic gradients from Lemma 54, as well as the fact that $\max_{\theta \in \Theta^\Phi} c^{\eta,t}(\theta) \geq c^{\eta,t}(\theta^*_{t+1})$ and since $\Phi\theta^*_{t+1}$ is feasible, we have $\|(\Phi\theta^*_{t+1})^\top (P - B)\|_1 = 0$ and $\|[\Phi\theta^*_{t+1}]_{(\delta,-)}\|_1 = 0$. □

We remark that we did not relax the constraint $(\Phi\theta)^\top 1 = 1$ and in fact when we use

Projected Gradient Ascent we are projecting onto a subset of that hyperplane, although Φ has $|S||A|$ rows we can precompute the vector $\Phi^\top 1 \in \mathbb{R}^d$ so that all projections to the subset of the hyper plane given by $(\Phi\theta)^\top 1 = 1$ can be done in $O(poly(d))$ time.

The next lemma bounds the largest difference the function $F_t(\Phi\theta)$ can take over $\theta \in \Theta^\Phi$. It will be clear later why this bound is needed.

Lemma 61. *For all $t = 1, ..., T$. It holds that*

$$\max_{\theta_1,\theta_2\in\Theta^\Phi} F_t(\Phi\theta_1) - F_t(\Phi\theta_2) \leq t\left[2 + \frac{1}{\eta}\ln(|S||A|)\right].$$

Proof. By definition of F_t it suffices to bound

$$\sum_{i=1}^{t}\langle r_i, \Phi\theta_1 - \Phi\theta_2\rangle + \frac{t}{\eta}\left[R^\delta(\Phi\theta_2) - R^\delta(\Phi\theta_1)\right].$$

Now, we have

$$\begin{aligned}
&\sum_{i=1}^{t}\langle r_i, \Phi\theta_1 - \Phi\theta_2\rangle \leq \sum_{i=1}^{t}\|r_i\|_\infty\|\Phi\theta_1 - \Phi\theta_2\|_1 \quad \text{By Cauchy-Schwarz}\\
&\leq \sum_{i=1}^{t} 1\|\Phi\theta_1 - \Phi\theta_2\|_1\\
&\leq \sum_{i=1}^{t}\|\Phi\theta_1\|_1 + \|\Phi\theta_2\|_1 \quad \text{by triangle inequality}\\
&\leq 2t,
\end{aligned}$$

where the last inequality holds since all entries of Φ and θ are nonnegative, and $(\Phi\theta)^\top 1 = 1$ for all $\theta \in \Theta^\Phi$.

It is well known that the minimizer of $R(\mu)$ for $\mu \in \Delta^{|S||A|}$ is $-\ln(|S||A|)$. Moreover, its optimal solution μ^* is equal to the vector with value $1/(|S||A|)$ on each of its entries, which is of course in the interior of the simplex. Notice that since R^δ is an extension of R, if δ is sufficiently small (which we ensure by the choice of δ later in the analysis), the

minimizer of $R^\delta(\Phi\theta)$ for $\theta \in \Theta^\Phi$ will be bounded below by $-\ln(|S||A|)$. That is

$$-\ln(|S||A|) \leq \min_{\theta \in \Theta^\Phi} R^\delta(\Phi\theta).$$

We upper bound $\max_{\theta\in\Theta^\Phi} R^\delta(\Phi\theta)$. By construction $R^\delta(\Phi\theta) \leq R(\Phi\theta)$ for all $0 \leq \theta \in \Theta^\Phi$. Since $\theta \geq 0, 1^\top \Phi\theta = 1$ defines the set Φ^Θ and Φ has probability distributions as its columns it holds that $R(\Phi\theta) \leq 0$, thus $R^\delta(\Phi\theta) \leq 0$. We have shown that

$$\sum_{i=1}^{t} \langle r_i, \Phi\theta_1 - \Phi\theta_2 \rangle + \frac{t}{\eta}\left[R^\delta(\Phi\theta_2) - R^\delta(\Phi\theta_1)\right] \leq 2t + \frac{t}{\eta}\left[\ln(|S||A|)\right]$$

which finishes the proof. □

Lemma 59 assumes we have at our disposal a vector $\Phi\tilde{\theta}_{t+1}$ such that $\|[\Phi\tilde{\theta}_{t+1}]_{(\delta,-)}\|_1 \leq \epsilon'$ and $\|(\Phi\tilde{\theta}_{t+1})^\top (P - B)\|_1 \leq \epsilon''$, and $F_t(\Phi\theta^*_{t+1}) - F_t(\Phi\tilde{\theta}_{t+1}) \leq \epsilon'''$ for some $\epsilon', \epsilon'', \epsilon''' \geq 0$. We now show how to obtain such error bounds by running at each time step t, $K(t)$ iterations of PSGA and using Lemma 60.

Lemma 62. *For $t = 1, ..., T$, let $b_{K(t)}$ the right hand side of the equation in the bound of Lemma 60 and assume the same conditions hold. After $K(t)$ iterations of PSGA, with probability at least $1 - \kappa$, it holds that*

$$\begin{aligned}
\|[\Phi\tilde{\theta}_{t+1}]_{(\delta,-)}\|_1 &\leq \frac{1}{H_t}\left[b_{K(t)} + t[2 + \frac{1}{\eta}\ln(|S||A|)]\right], \\
\|(\Phi\tilde{\theta}_{t+1})^\top (P - B)\|_1 &\leq \frac{1}{H_t}\left[b_{K(t)} + t[2 + \frac{1}{\eta}\ln(|S||A|)]\right], \\
F_t(\Phi\theta^*_{t+1}) - F_t(\Phi\tilde{\theta}_{t+1}) &\leq b_{K(t)}.
\end{aligned}$$

Proof. To show the first two inequalities, notice that Lemma 60 implies

$$\begin{aligned}
H_t\|[\Phi\tilde{\theta}_{t+1}]_{(\delta,-)} + H_t\|(\Phi\tilde{\theta}_{t+1})^\top (P - B)\|_1 &\leq b_{K(t)} + F_t(\Phi\tilde{\theta}_{t+1}) - F_t(\Phi\theta^*_{t+1}) \\
&\leq b_{K(t)} + t\left[2 + \frac{1}{\eta}\ln(|S||A|)\right],
\end{aligned}$$

where the last inequality holds by Lemma 61. Since $\|\cdot\|_1 \geq 0$ we get the desired results. To show that $F_t(\Phi\theta^*_{t+1}) - F_t(\Phi\tilde{\theta}_{t+1}) \leq b_{K(t)}$ again use Lemma 60 and the fact that $\|\cdot\|_1 \geq 0$.

□

We are ready to prove the main theorem from this section.

Proof of Theorem 28. Recall the Φ-MDP-Regret regret bound from Equation 5.11.

$$\begin{aligned}
&\max_{\pi \in \Pi^\Phi} R(\pi, T) \\
&\leq \mathbb{E}_{PSGA}[(4\tau + 4) \\
&+ [\max_{\mu \in \Delta^\Phi_{M,\delta}} \sum_{t=1}^{T} \langle \mu, r_t \rangle - \sum_{t=1}^{T} \langle \mu^{\Phi\tilde{\theta}_t}, r_t \rangle] + [\sum_{t=1}^{T} \sum_{i=0}^{t-i} e^{-\frac{i}{\tau}} \|\mu^{\Phi\tilde{\theta}_{t-i}} - \mu^{\Phi\tilde{\theta}_{t-(i+1)}}\|_1]].
\end{aligned}$$

Since it is cumbersome to work with the $\mathbb{E}_{PSGA}[\cdot]$ in our bounds let us make the following argument. For $t = 1, ..., T$, define $\mathcal{E}_t$ be the event that the inequality in Lemma 60 holds, let $\mathcal{E} \triangleq \cap_{t=1}^{T} \mathcal{E}_t$. For any random variable X we know that $\mathbb{E}_{PSGA}[X] = \mathbb{E}_{PSGA}[X|\mathcal{E}]P(\mathcal{E}) + \mathbb{E}_{PSGA}[X|\mathcal{E}^c]P(\mathcal{E}^c)$. Let us work conditioned on the event $\mathcal{E}$, we will later bound $\mathbb{E}_{PSGA}[X|\mathcal{E}^c]P(\mathcal{E}^c)$.

By triangle inequality, Cauchy-Schwarz inequality, and the fact $\|r_t\|_\infty \leq 1$ for $t = 1, ..., T$, it holds that

$$\begin{aligned}
\max_{\mu \in \Delta^\Phi_{M,\delta}} \sum_{t=1}^{T} \langle \mu, r_t \rangle - \sum_{t=1}^{T} \langle \mu^{\Phi\tilde{\theta}_t}, r_t \rangle \leq \max_{\mu \in \Delta^\Phi_{M,\delta}} \sum_{t=1}^{T} \langle \mu, r_t \rangle - \sum_{t=1}^{T} \langle \mu^{\Phi\theta^*_t}, r_t \rangle \\
+ \sum_{t=1}^{T} \|\mu^{\Phi\theta^*_t} - \mu^{\Phi\tilde{\theta}_t}\|_1.
\end{aligned}$$

Notice that

$$\begin{aligned}
&\|\mu^{\Phi\tilde{\theta}_{t-i}} - \mu^{\Phi\tilde{\theta}_{t-(i+1)}}\|_1 \\
\leq &\|\mu^{\Phi\theta^*_{t-i}} - \mu^{\Phi\theta^*_{t-(i+1)}}\|_1 + \|\mu^{\Phi\tilde{\theta}_{t-i}} - \mu^{\Phi\theta^*_{t-i}}\|_1 + \|\mu^{\Phi\tilde{\theta}_{t-(i+1)}} - \mu^{\Phi\theta^*_{t-(i+1)}}\|_1.
\end{aligned}$$

Therefore, we have

$$
\begin{aligned}
&\max_{\pi \in \Pi^{\Phi}} R(\pi, T) \\
&\leq 2(2\tau + 2) + \left[\max_{\mu \in \Delta^{\Phi}_{M,\delta}} \sum_{t=1}^{T} \langle \mu, r_t \rangle - \sum_{t=1}^{T} \langle \mu^{\Phi\theta^*_t}, r_t \rangle \right] \\
&+ \left[\sum_{t=1}^{T} \sum_{i=0}^{t-i} e^{-\frac{i}{\tau}} \| \mu^{\Phi\theta^*_{t-i}} - \mu^{\Phi\theta^*_{t-(i+1)}} \|_1 \right] \\
&+ \sum_{t=1}^{T} \| \mu^{\Phi\theta^*_t} - \mu^{\Phi\tilde{\theta}_t} \|_1 + \sum_{t=1}^{T} \sum_{i=0}^{t-i} e^{-\frac{i}{\tau}} \left(\| \mu^{\Phi\tilde{\theta}_{t-i}} - \mu^{\Phi\theta^*_{t-i}} \|_1 + \| \mu^{\Phi\tilde{\theta}_{t-(i+1)}} - \mu^{\Phi\theta^*_{t-(i+1)}} \|_1 \right) \\
&\leq O\left(\tau + 4\sqrt{\tau T} \ln(T) + \sqrt{\tau T} \ln(|S||A|) + e^{-\sqrt{T}} T |S||A| \right) \\
&+ \sum_{t=1}^{T} \| \mu^{\Phi\theta^*_t} - \mu^{\Phi\tilde{\theta}_t} \|_1 + \sum_{t=1}^{T} \sum_{i=0}^{t-i} e^{-\frac{i}{\tau}} \left(\| \mu^{\Phi\tilde{\theta}_{t-i}} - \mu^{\Phi\theta^*_{t-i}} \|_1 + \| \mu^{\Phi\tilde{\theta}_{t-(i+1)}} - \mu^{\Phi\theta^*_{t-(i+1)}} \|_1 \right),
\end{aligned}
$$

where the second inequality follows from the proof of Theorem 27 since we chose the same parameters $\eta = \sqrt{\frac{T}{\tau}}, \delta = e^{-\sqrt{T}}$.

If we choose $K(t)$ such that $\| \mu^{\Phi\theta^*_t} - \mu^{\Phi\tilde{\theta}_t} \|_1$ are less than or equal to a constant $\epsilon(\epsilon'_t, \epsilon''_t, \epsilon'''_t, K(t))$ for all $t = 1, ..., T$ we have

$$
\begin{aligned}
&\sum_{t=1}^{T} \| \mu^{\Phi\theta^*_t} - \mu^{\Phi\tilde{\theta}_t} \|_1 + \sum_{t=1}^{T} \sum_{i=0}^{t-i} e^{-\frac{i}{\tau}} \left(\| \mu^{\Phi\tilde{\theta}_{t-i}} - \mu^{\Phi\theta^*_{t-i}} \|_1 + \| \mu^{\Phi\tilde{\theta}_{t-(i+1)}} - \mu^{\Phi\theta^*_{t-(i+1)}} \|_1 \right) \\
&\leq T\epsilon + 2T\epsilon(1 + \int_0^{\infty} e^{-\frac{x}{\tau}} dx) \\
&\leq T\epsilon + 2T\epsilon(1 + \tau) \\
&= T(1 + 2(1 + \tau))\epsilon.
\end{aligned}
$$

We have that

$$
\max_{\pi \in \Pi^{\Phi}} R(\pi, T) \leq O\left(\tau + 4\sqrt{\tau T} \ln(T) + \sqrt{\tau T} \ln(|S||A|) + e^{-\sqrt{T}} T |S||A| + T\tau\epsilon \right).
$$

Let $\epsilon'_t = \epsilon''_t = \frac{1}{H_t} \left[b_{K(t)} + t[2 + \frac{1}{\eta} \ln(|S||A|)] \right]$, $\epsilon'''_t = b_{K(t)}$. By Lemma 59, we have

that

$$\epsilon \leq \tau \ln(\frac{1}{\epsilon'})(2\epsilon' + \epsilon'') + 3\epsilon' + c(\epsilon' + \epsilon'') + \sqrt{\frac{2\eta}{t}(\epsilon''' + G_{F_t} c(\epsilon' + \epsilon''))}.$$

By Lemma 58 we know that $G_{F_t} \leq t(1 + 2\sqrt{\tau}\ln(dW))$ so that

$$\epsilon \leq \tau \ln(\frac{1}{\epsilon'})(2\epsilon' + \epsilon'') + 3\epsilon' + c(\epsilon' + \epsilon'') + \sqrt{\frac{2\sqrt{T}}{\sqrt{\tau}}(\epsilon''' + c[1 + 2\sqrt{\tau}\ln(dW)](\epsilon' + \epsilon''))},$$

where we plugged in the value for η. It is easy to see that the right hand side of the last inequality bounded above by $O(\tau \ln(\frac{1}{\epsilon'})T^{1/4}c\sqrt{dW}(\epsilon' + \epsilon'' + \epsilon'''))$. So that forcing all $\epsilon', \epsilon'', \epsilon'''$ to be $O(\frac{1}{\sqrt{dW}\tau^{3/2}T^{3/4}})$ will ensure $T\tau\epsilon$ to be $O(c\sqrt{\tau T})$ ensuring that $\max_{\pi \in \Pi^\Phi} R(\pi, T) \leq O(c\sqrt{\tau T}\ln(T)\ln(|S||A|))$.

Since $\epsilon' = \epsilon'' = \frac{1}{H_t}b_{K(t)} + \frac{1}{H_t}t2 + \frac{1}{H_t}t\frac{\sqrt{\tau}}{\sqrt{T}}\ln(|S||A|)$ we choose $H_t = \sqrt{dW}t\tau^2T^{3/4}$, this ensures that $\frac{1}{H_t}t2 + \frac{1}{H_t}t\frac{\sqrt{\tau}}{\sqrt{T}}\ln(|S||A|)$ are bounded above by $O(\frac{1}{\sqrt{dW}\tau^{3/2}T^{3/4}})$. We now must choose $K(t)$ so that $\frac{1}{H_t}b_{K(t)}$ and ϵ_t''' are both $O(\frac{1}{\sqrt{dW}\tau^{3/2}T^{3/4}})$. Since by the choice of H_t we have $\frac{1}{H_t}b_{K(t)} \leq b_{K(t)}$ it suffices to bound $b_{K(t)}$.

Set $\kappa = \frac{1}{T^2}$ in Lemma 60 and recall we are working conditioned on $\mathcal{E}$, we have that for all $t = 1, ..., T$

$$\begin{aligned}
b_{K(t)} &= \frac{\sqrt{d}Wt\sqrt{d} + H_t(C_1 + C_2) + \frac{t}{\eta}(1 + \ln(Wd) + |\ln(\delta)|)C_1}{\sqrt{K(t)}} \\
&+ \sqrt{\frac{(1 + 4dW^2K(t))(2\ln(\frac{1}{\kappa}) + d\ln(1 + \frac{dW^2K(t)}{d}))}{K(t)^2}} \\
&\leq O(\frac{WtdH_t(C_1 + C_2)\sqrt{T}\sqrt{\tau}\ln(WTd)}{\sqrt{T}\sqrt{K(t)}}) \\
&= O(\frac{Wt^2d(C_1 + C_2)\tau^{5/2}T^{3/4}\ln(WTd)}{\sqrt{K(t)}}).
\end{aligned}$$

Setting

$$\frac{Wt^2d(C_1+C_2)\tau^{5/2}T^{3/4}\ln(WTd)}{\sqrt{K(t)}} = \frac{1}{\sqrt{dW}\tau^{3/2}T^{3/4}}$$

and solving for $K(t)$, we get that $K(t) = \left[W^{3/2}t^2d^{3/2}\tau^4(C_1+C_2)T^{3/2}\ln(WTd)\right]^2$, which ensures $b_{K(t)} = O(\frac{1}{\sqrt{dW}\tau^{3/2}T^{3/4}})$.

By the choice of κ in Lemma 60, we have that for each $t = 1, ..., T$ with probability at least $1 - \frac{1}{T^2}$, $\|\mu^{\Phi\theta_t^*} - \mu^{\Phi\tilde{\theta}_t}\|_1 \leq O(\sqrt{dW}\frac{1}{\tau^{3/2}T^{3/4}})$. This implies that

Φ-MDP-Regret

$$\leq O(c\sqrt{\tau T}\ln(T)\ln(|S||A|))P(\mathcal{E})$$
$$+\left[O\left(\tau + 4\sqrt{\tau T}\ln(T) + \sqrt{\tau T}\ln(|S||A|) + e^{-\sqrt{T}}T|S||A| + \sum_{t=1}^{T}\|\mu^{\Phi\theta_t^*} - \mu^{\Phi\tilde{\theta}_t}\|_1\right)\right]P(\mathcal{E}^c)$$

Notice that since $\mu^{\Phi\theta_t^*}$, and $\mu^{\Phi\tilde{\theta}_t}$ are probability distributions then $\|\mu^{\Phi\theta_t^*} - \mu^{\Phi\tilde{\theta}_t}\|_1 \leq 2$. So that

$$\Phi\text{-MDP-Regret} \leq O(c\sqrt{\tau T}\ln(T)\ln(|S||A|)) + O(T)P(\mathcal{E}^c)$$

where we upper bounded $P(\mathcal{E})$ with 1. Notice that by the choice of κ,
$P(\mathcal{E}^c) = P(\cup_{t=1}^{T}\mathcal{E}_i^c) \leq \sum_{t=1}^{T}P(\mathcal{E}_t^c) \leq \frac{1}{T}$ so that $O(T)P(\mathcal{E}^c) = O(1)$. This completes the proof.

□

5.8 Conclusion

We consider Markov Decision Processes (MDPs) where the transition probabilities are known but the rewards are unknown and may change in an adversarial manner. We provide a simple online algorithm, which applies Regularized Follow the Leader (RFTL) to the

linear programming formulation of the average reward MDP. The algorithm achieves a regret bound of $O(\sqrt{\tau(\ln|S| + \ln|A|)T}\ln(T))$, where S is the state space, A is the action space, τ is the mixing time of the MDP, and T is the number of periods. The algorithm's computational complexity is polynomial in $|S|$ and $|A|$ per period.

We then consider a setting often encountered in practice, where the state space of the MDP is too large to allow for exact solutions. We approximate the state-action occupancy measures with a linear architecture of dimension $d \ll |S||A|$. We then propose an approximate algorithm that relaxes the constraints in the RFTL algorithm, and then solve the relaxed problem using stochastic gradient descent method. The computational complexity of this approximate algorithm is independent of the size of state space $|S|$ and the size of action space $|A|$. We prove a regret bound of $O(c_{S,A}\ln(|S||A|)\sqrt{\tau T}\ln(T))$ compared to the best static policy approximated by the linear architecture, where $c_{S,A}$ is a problem dependent constant. To the best of our knowledge, this is the first $\tilde{O}(\sqrt{T})$ regret bound for large scale MDPs with changing rewards.

Appendices

APPENDIX A

RISK-AVERSE CONVEX BANDIT

A.1 More Preliminaries

A.1.1 Some Useful Concentration Results

In this section we present results on how quickly random functions uniformly concentrate around their mean.

Lemma 63. *[118][Theorem 5] Let* $\hat{F}(x) = \frac{1}{N}\sum_{n=1}^{N} f(x, \xi_n)$ *where* $f(\cdot, \xi)$ *is* L*-Lipschitz with function values bounded by* R *and the set where it is defined has diameter* B*. Let* $F(x) := \mathbb{E}_{\xi}[f(x, \xi)]$*. Then*

$$P(\sup_{x \in X} |F(x) - \hat{F}(x)| \geq \epsilon) \leq O(d^2(\frac{LB}{\epsilon})^d \exp(-\frac{N\epsilon^2}{128LR})). \tag{A.1}$$

This result implies the following two lemmas.

Lemma 64. *With probability at least* $1 - \delta$*, for any* $x \in X$*, over a sample size* N

$$|F(x) - \hat{F}(x)| \leq \tilde{O}(\sqrt{\frac{LRd\ln(\frac{1}{\delta})}{N}}).$$

Proof. Setting the right hand side of (A.1) equal to δ and solving for ϵ gives

$$\epsilon = \sqrt{\frac{128LR[2\ln(\frac{d}{\sqrt{\delta}}) + d\ln(LB) + d\ln(\frac{1}{\epsilon})]}{N}}$$

Since we must bound ϵ by above, we now bound $\ln(\frac{1}{\epsilon})$. Using the previous equality we

have

$$\ln(\frac{1}{\epsilon}) = \frac{1}{2}\ln(\frac{N}{128LR[2\ln(\frac{d}{\sqrt{\delta}}) + d\ln(\frac{LB}{\epsilon})]})$$

since $\ln(\frac{LB}{\epsilon})$ is large and in the denominator, we have

$$\ln(\frac{1}{\epsilon}) \leq \frac{1}{2}\ln(\frac{N}{256LR\ln(\frac{d}{\sqrt{\delta}})})$$

this implies[1]

$$\begin{aligned}
\epsilon &\leq \sqrt{\frac{128LR[2\ln(\frac{d}{\sqrt{\delta}}) + d\ln(LB) + d\frac{1}{2}\ln(\frac{N}{256LR\ln(\frac{d}{\sqrt{\delta}})})]}{N}} \\
&= \sqrt{\frac{\kappa LRd\ln(\frac{dLBN}{\sqrt{\delta}256LR\ln(\frac{d}{\delta})})}{N}} \\
&= \tilde{O}(\sqrt{\frac{LRd\ln(\frac{1}{\delta})}{N}})
\end{aligned}$$

□

Lemma 65.

$$\mathbb{E}[\sup_{x\in X}|F(x) - \hat{F}(x)|] \leq \tilde{O}(\frac{\sqrt{LRd}}{\sqrt{N}})$$

Proof. Recall that for a nonnegative random variable X it holds that $\mathbb{E}[X] = \int_0^\infty P(X > t)dt$. We have from (A.1)

$$\begin{aligned}
P(\sup_{x\in Z}|F(x) - \hat{F}(x)| > \epsilon) &\leq O(d^2(\frac{LB}{\epsilon})^d \exp(-\frac{N\epsilon^2}{128LR})) \\
&= \exp[-(\frac{N\epsilon^2}{128LR} + d\ln(\epsilon) - 2\ln(d) - d\ln(LB))]
\end{aligned}$$

Let $\lambda(\epsilon) = a\epsilon^2 + d\ln(\epsilon)$ with $a := \frac{N}{128LR}$ and notice that when $\epsilon \geq \sqrt{\frac{d}{2a}}$ the second derivative of $\lambda(\cdot)$ is nonnegative and therefore the function is convex in that domain thus

[1]Throughout the section we let κ be some universal constant that may change from line to line.

we can lower bound it with its first order Taylor approximation at $\sqrt{\frac{d}{2a}}$.

$$\lambda(\epsilon) \geq 2\sqrt{2ad}\epsilon - 2d + \frac{d}{2} + \frac{d}{2}\ln(\frac{d}{2a})$$

Therefore, for $\epsilon \geq \sqrt{\frac{d}{2a}}$

$$\begin{aligned}
P(\sup_{x\in Z}|F(x) - \hat{F}(x)| > \epsilon) &\leq \exp[-(2\sqrt{2ad}\epsilon - 2d + \frac{d}{2} + \frac{d}{2}\ln(\frac{d}{2a}) - 2\ln(d) - d\ln(LB))] \\
&\leq \exp[-(2\sqrt{2ad}\epsilon - 2d + \frac{d}{2}\ln(\frac{d}{2a}) - 2\ln(d) - d\ln(LB))] \\
&\leq \exp[-(2\sqrt{2ad}\epsilon - 2d - \frac{d}{2}\ln(2a) - 2\ln(d) - d\ln(LB))] \\
&= \exp[-(2\sqrt{2ad}\epsilon) + \theta]
\end{aligned}$$

where $\theta := 2d + \frac{d}{2}\ln(2a) + 2\ln(d) + d\ln(LB)$. We have

$$\begin{aligned}
\mathbb{E}[\sup_{x\in X}|F(x) - \hat{F}(x)|] &\leq \int_0^\infty \min[1, \exp[-(2\sqrt{2ad}\epsilon) + \theta]]d\epsilon \\
&= \int_0^{\epsilon'} d\epsilon + \int_{\epsilon'}^\infty \exp[-2\sqrt{2ad}\epsilon + \theta]d\epsilon \quad \epsilon' = \frac{\theta}{2\sqrt{2ad}} \\
&= \epsilon' + \frac{\exp[\theta - 2\sqrt{2ad}\epsilon']}{2\sqrt{2ad}} \\
&= \frac{1}{2\sqrt{2ad}}[\theta + 1] \\
&= \frac{\sqrt{128LR}}{2\sqrt{2dN}}[2d + \frac{d}{2}\ln(2a) + 2\ln(d) + d\ln(LB) + 1] \\
&= \tilde{O}(\frac{\sqrt{LRd}}{\sqrt{N}})
\end{aligned}$$

□

A.1.2 Conditional Value at Risk

Lemma 66. *Let ξ be a random variable supported in Ξ with probability distribution P. Let $f : X \times \Xi \to \mathbb{R}$ and assume $0 \leq f(x, \xi) \leq 1$ for all $x \in X$ and $\xi \in \Xi$. If $f(\cdot, \xi)$ is*

G-Lipschitz then so is $CVaR_\alpha[F](x)$.

Proof. By Theorem 6.4 in [119] for any $x \in X$. We have

$$CVaR_\alpha[F](x) = \sup_{\xi \in \Theta} \mathbb{E}_\xi[f(x, \xi)]$$

where Θ is some family of probability distributions.

Since convex combinations of G-Lipschitz functions is G-Lipschitz we have that for any $x_1 \in X$

$$\mathbb{E}_{\xi \in \Theta_1^*}[f(x_1, \xi)] - \mathbb{E}_{\xi \in \Theta_1^*}[f(x_2, \xi)] \leq G||x_1 - x_2||$$

where Θ_1^* is the probability distribution that maximizes $\mathbb{E}_{\xi \in \Theta}[f(x_1, \xi)]$ (assuming it exists). Since

$$\mathbb{E}_{\xi \in \Theta_1^*}[f(x_1, \xi)] - \mathbb{E}_{\xi \in \Theta_2^*}[f(x_2, \xi)] \leq \mathbb{E}_{\xi \in \Theta_1^*}[f(x_1, \xi)] - \mathbb{E}_{\xi \in \Theta_1^*}[f(x_2, \xi)]$$

by combining the two inequalities we have

$$CVaR_\alpha[F](x_1) - CVaR_\alpha[F](x_2) \leq G||x_1 - x_2||$$

a symmetry argument yields the other side of the inequality, this concludes the proof. □

Lemma 67. *Let X be a convex set with diameter $D_{||\cdot||}$ that contains the origin, that is for all $x_1, x_2 \in X$, $||x_1 - x_2|| \leq D_{||\cdot||}$. Let $X_\delta := \{x : x \in (1 - \delta)X\}$. For any $x \in X$ let $x_\delta := \Pi_{X_\delta}(x)$ where the projection is taken with respect to any norm $||\cdot||$. Then*

$$||x - x_\delta|| \leq \delta D_{||\cdot||} \tag{A.2}$$

Proof. Notice $(1-\delta)x \in X_\delta$

$$\begin{aligned} ||x - x_\delta|| &\leq ||x - (1-\delta)x|| \quad \text{By definition of } \Pi \\ &\leq \delta ||x|| \\ &\leq \delta D_{||\cdot||} \quad \text{since } X \text{ contains the origin} \end{aligned}$$

□

Lemma 68. *Let* $x = [x_1, x_2]^\top$. *Define* $||x|| = ||x_1||_2 + ||x_2||_\infty$. *Then*

$$||x||_* = \max\{||x_1||_2, ||x_2||_1\}$$

Proof. By definition of dual norm we have

$$\begin{aligned} ||x||_* &= \max_{||y|| \leq 1} x_1^\top y_1 + x_2^\top y_2 \\ &= \max_{||y_1||_2 + ||y_2||_\infty \leq 1} x^\top y \\ &= \max_{c_1 + c_2 \leq 1} c_1 ||x_1||_2 + c_2 ||x_2||_1 \\ &= \max\{||x_1||_2, ||x_2||_1\} \end{aligned}$$

□

A.2 Analysis of Algorithm 3

The following algorithm, a generalization of Algorithm 1, will guarantee vanishing $\bar{\mathcal{R}}_T^\rho$ and $\mathcal{R}_T^\rho$ by exploiting the Kusuoka representation of risk measure ρ.

Notice that due to Lemma 3, $g_t := [g_t^1; g_t^2]$ is a one point gradient estimator of the smoothened version of $\mathcal{G}$, $\hat{\mathcal{G}}$.

The proofs of Theorems 8 and 9 will be similar to that of Theorems 2 and 3, however we must be careful to make sure we do not introduce unnecessary factors of N, d and $\frac{1}{\alpha}$.

Algorithm 3

Input: $X \subset \mathbb{R}^d$, $x_1 \in X$, $z_1 \in Z$ step size η, δ
for $t = 1, ..., T$ **do**
  Sample $u \sim \mathbb{S}^{d+N}$
  Let $u_t^1 = [u_1; ...; u_d]$ and $u_t^2 = [u_{d+1}; ...; u_{d+N}]$
  Play $\tilde{x}_t := x_t + \delta u^1$, observe $f_t(\tilde{x}_t)$
  Let $\tilde{z}_t = z_t + \delta u^2$
  Let $g_t^1 := \frac{(d+N)}{\delta}(\mathcal{G}_t(\tilde{x}_t, \tilde{z}_t))u_t^1$
  Let $g_t^2 := \frac{(d+N)}{\delta}(\mathcal{G}_t(\tilde{x}_t, \tilde{z}_t))u_t^2$
  Update $x_{t+1} \leftarrow \Pi_{X_\delta}(x_t - \eta g_t^1)$
  Update $z_{t+1} \leftarrow \Pi_{Z_\delta}(z_t - \eta g_t^2)$
end for

Lemma 69. $||\nabla \mathcal{G}|| \leq N(G+1)+1$

Proof.

$$
\begin{aligned}
||\nabla \mathcal{G}|| &= \sqrt{\sum_{i=1}^{d}(\sum_{n=1}^{N}\mu_n \nabla_{x_i}\mathcal{L}_n)^2 + \sum_{n=1}^{N}(\mu_n \nabla_{z_n}\mathcal{L}_n)^2} \\
&\leq \sqrt{\sum_{i=1}^{d}(||\mu||_1||\nabla_{x_i}\mathcal{L}_n||_\infty)^2 + \sum_{n=1}^{N}(\mu_n \nabla_{z_n}\mathcal{L}_n)^2} \quad ||.||_\infty \text{ is over n=1,...,N} \\
&\leq \sqrt{\sum_{i=1}^{d}||\nabla_{x_i}\mathcal{L}_n||_\infty^2 + \sum_{n=1}^{N}\mu_n \nabla_{z_n}\mathcal{L}_n^2} \quad \text{since } \sum_{n=1}^{N}\mu_n = 1, \text{ and } \mu_i \leq 1 \\
&\leq \sqrt{\sum_{i=1}^{d}||\nabla_{x_i}\mathcal{L}_n||_\infty^2 + \sum_{n=1}^{N}\mu_n (1+N)^2} \\
&\leq \sqrt{\sum_{i=1}^{d}||\nabla_{x_i}\mathcal{L}_n||_\infty^2} + \sqrt{\sum_{n=1}^{N}\mu_n (1+N)^2} \\
&\leq \sqrt{\sum_{i=1}^{d}||N\nabla_{x_i} f||_\infty^2} + \sqrt{\sum_{n=1}^{N}\mu_n (1+N)^2} \\
&\leq NG + (1+N) \quad \text{since } \sum_{n=1}^{N}\mu_n = 1
\end{aligned}
$$

□

Lemma 70. *Running online gradient descent on* $\{\mathcal{G}_t\}_{t=1}^T$ *ensures that for all* $x \in X$ *and*

all $z \in Z$

$$2[\sum_{t=1}^{T} \mathcal{G}_t(x_t, z_t) - \sum_{t=1}^{T} \mathcal{G}_t(x, z)] \leq \frac{||x_T - x^*||^2 + \sum_{n=1}^{d} \mu_n ||z_{t,n} - z_n^*||^2}{\eta} + \eta[\sum_{t=1}^{T}(||\nabla_x \mathcal{G}_t(x_t, y_t) + \sum_{n=1}^{N} \mu_n |\nabla_{z_n} \mathcal{L}(x_t, z_t)|^2)].$$

Proof.

$$\begin{aligned}
&2[\sum_{t=1}^{T} \mathcal{G}_t(x_t, z_t) - \sum_{t=1}^{T} \mathcal{G}_t(x, z)] \\
&\leq 2 \sum_{t=1}^{T} \nabla \mathcal{G}_t(x_t, z_t)^\top ([x_t; z_t] - [x; z]) \\
&= 2 \sum_{t=1}^{T} \nabla_x \mathcal{G}_t(x_t, z_t)^\top (x_t - x) + 2 \sum_{t=1}^{T} \sum_{n=1}^{d} \mu_n \nabla_z \mathcal{L}(x_t, z_t)(z_{t,n}.z_n) \\
&\leq \frac{||x_T - x||^2}{\eta} + \sum_{n=1}^{N} \mu_n \frac{||z_{T,n} - z_n||^2}{\eta} + \eta[\sum_{t=1}^{T}(||\nabla_x \mathcal{G}_t|| + \sum_{n=1}^{d} \mu_n (\nabla_z \mathcal{L}_{t,n})^2)]
\end{aligned}$$

by Equations 2.7 and 2.8

□

Lemma 71. *Let* $y^* = (x^*, z^*) \in \arg\min \mathbb{E}_\xi[\sum_{t=1}^{T} \mathcal{G}_t(x, z)]$. *With appropriate choice of parameters* η, δ *we have*

$$\mathbb{E}_{int}[\sum_{t=1}^{T} \mathcal{G}_t(\tilde{y}_t)] - \sum_{t=1}^{T} \mathcal{G}_t(y^*) \leq O(dN^{3/2}T^{3/4})$$

Proof. First we need a bound on $\sum_{t=1}^{T} \mathcal{G}_t(y_\delta^*) - \sum_{t=1}^{T} \mathcal{G}_t(y^*)$, where $y_\delta^* = \Pi_{X_\delta \times Z_\delta}(y^*)$. If $\mathcal{G}$ is Lipschitz L with respect to some norm $||\cdot||$, by Lemma 1 we have $||\nabla \mathcal{G}||_* \leq L$. For any $y = [x; z]$ with $x \in X$ and $z \in Z$, let us use $||y|| = ||x||_2 + ||z||_\infty$ with dual norm

$||y||_* = \max\{||x||_2, ||z||_1\}$ (see Lemma 68 in the Appendix).

$$\begin{aligned}\sum_{t=1}^{T}\mathcal{G}_t(y_\delta^*) - \sum_{t=1}^{T}\mathcal{G}_t(y^*) &\leq TL||y^* - y_\delta^*|| \\ &\leq \delta TLD_{\mathcal{G}}^{||\cdot||} \quad \text{by Lemma 67 in the Appendix} \\ &\leq O(\delta TGN).\end{aligned}$$

The last inequality holds because of the following two facts,
1) $||\nabla\mathcal{G}||_* = \max\{||\nabla_x\mathcal{G}||_2, ||\nabla_z\mathcal{G}||_1\} \leq \max\{G, \sum_{n=1}^{N}\mu[1+N]\} \leq G+1+N$ and 2) $||y_1 - y_2|| = ||x_1 - x_2||_2 + ||z_1 - z_2||_\infty \leq D_X + 2 := D_{\mathcal{G}}^{||\cdot||}$. Let $\mathbb{E}_{int}$ be the expectation taken with respect to the internal randomization of the algorithm. Following the proof of Lemma 8 we have

$$\begin{aligned}&\mathbb{E}_{int}[\sum_{t=1}^{T}\mathcal{G}_t(\tilde{y}_t)] - \sum_{t=1}^{T}\mathcal{G}_t(y^*) \\ &\leq \mathbb{E}_{int}[\sum_{t=1}^{T}\mathcal{G}_t(y_t)] - \sum_{t=1}^{T}\mathcal{G}_t(y^*) + \delta G_{\mathcal{G}}T \quad \mathcal{G} \text{ is } G_{\mathcal{G}}\text{-Lipschitz and } ||y-\tilde{y}|| \leq \delta \\ &\leq \mathbb{E}_{int}[\sum_{t=1}^{T}\mathcal{G}_t(y_t)] - \sum_{t=1}^{T}\mathcal{G}_t(y_\delta^*) + \delta G_{\mathcal{G}}T + O(\delta TGN) \\ &\leq \mathbb{E}_{int}[\sum_{t=1}^{T}\hat{\mathcal{G}}_t(y_t)] - \sum_{t=1}^{T}\hat{\mathcal{G}}_t(y_\delta^*) + 3\delta G_{\mathcal{G}}T + \delta D_{\mathcal{G}}G_{\mathcal{G}}T \quad |\mathcal{G}(y) - \hat{\mathcal{G}}(y)| < \delta G_{\mathcal{G}} \\ &\leq \frac{||x_T - x^*||_2^2}{2\eta} + \sum_{n=1}^{N}\mu_n\frac{||z_{t,n} - z_n^*||_2^2}{2\eta} + \mathbb{E}_{int}[2\eta[\sum_{t=1}^{T}(||g_t^1||_2 + \sum_{n=1}^{d}\mu_n(g_{t,n}^2)^2)]] + 3\delta G_{\mathcal{G}}T \\ &\quad + O(\delta TGN) \quad \text{reduction to bandit feedback and Lemma 70} \\ &\leq \frac{D_X^2+2}{2\eta} + 2\eta\mathbb{E}_{int}[\sum_{t=1}^{T}(||g_t^1||_2^2 + \sum_{n=1}^{d}\mu_n(g_{t,n}^2)^2)] + 3\delta G_{\mathcal{G}}T + O(\delta TGN) \\ &\leq \frac{D_X^2+2}{2\eta} + 2\eta\frac{(d+N)^2N^2}{\delta^2}T + 3\delta G_{\mathcal{G}}T + O(\delta TGN) \\ &\leq O(dN^{3/2}T^{3/4})\end{aligned}$$

where we chose $\eta = O(\frac{1}{dN^{3/2}T^{3/4}})$ and $\delta = O(\frac{N^{1/2}}{T^{1/4}})$ and plugged in the bound on $G_{\mathcal{G}}$ from Lemma 69. □

Proof of Theorem 8. Take $\mathbb{E}_{\xi}[\cdot]$ on both sides of the result in Lemma 71 and interchange the expectations (this can be done using Fubini's Theorem and the uniform bound on $\mathcal{G}_t$). Noting that for all $x \in X$ and all $z \in [0,1]$ (in particular for every $(\tilde{x}_t, \tilde{z}_t)$) we have

$$\mathbb{E}_{\xi\sim P}[\mathcal{L}_n^t(x,z)] = z + \frac{1}{n/N}\mathbb{E}_{\xi\sim P}[f(x,\xi) - z]_+ \geq CVaR_{n/N}[F](x),$$

it follows that since $\mathcal{G}_t(x,z) := \sum_{n=1}^{N} \mu_n \mathcal{L}_n^t(x,z)$ we have $\mathbb{E}_{\xi\sim P}[\mathcal{G}_t(x,z)] \geq \rho[F](x)$. Noting that $\mathbb{E}_{\xi}[\sum_{t=1}^{T} \mathcal{G}_t(y^*)] = \min_{x\in X} \rho[F](x)$ we get the desired result. □

Proof of Theorem 9. We notice that strong convexity of $f(\cdot,\xi)$ implies strong convexity of $\rho[F](\xi)$ since each of the $C_{\alpha_i}[F](\cdot)$ in the Kusuoka representation of $\rho[F]$ is strongly convex. Let $x^* = argmin_{x\in X}\rho[F](x)$. We follow the proof of Theorem 3. Let the concentration error $CE = \rho[\{f_t(x^*)\}_{t=1}^T] - \min_{x\in X} \rho[\{f_t(x)\}_{t=1}^T]$.

$$\begin{aligned}
&\mathbb{E}[\rho[\{f_t(x_t)\}] - \min_{x\in X} \rho[\{f_t(x)\}]] \\
&= \mathbb{E}[\rho[\{f_t(x_t)\}] \pm \rho[\{f_t(x^*)\}] - \min_{x\in X} \rho[\{f_t(x)\}]] \\
&= \mathbb{E}[\sum_{n=1}^{N} \mu_n C_{n/N}[\{f_t(x_t)\}] - \rho[\{f_t(x^*)\}]] + \mathbb{E}[CE] \\
&\leq \mathbb{E}[\frac{N}{T}\sum_{t=1}^{T} |f_t(x_t) - f_t(x^*)|] + \mathbb{E}[CE] \quad \text{as in the last line of the proof of Theorem 3} \\
&\leq \frac{N}{T}\sum_{t=1}^{T} \mathbb{E}_t[||x_t - x^*||] + \mathbb{E}[CE] \\
&\leq \frac{N}{T}\sqrt{T}\sqrt{\sum_{t=1}^{T} \mathbb{E}_t[||x_t - x^*||^2]} + \mathbb{E}[CE] \\
&\leq \frac{N}{T}\sqrt{T}\sqrt{\sum_{t=1}^{T} \frac{2}{\beta}\mathbb{E}[\rho[F](x_t) - \rho[F](x^*)]} + \mathbb{E}[CE]
\end{aligned}$$

$$\leq O(\frac{d^{1/2}N^{7/4}}{\beta^{1/2}T^{1/8}}) + \mathbb{E}[CE] \quad \square$$

The expectation of the concentration error can be bounded as in the proof of Theorem 3 by $\tilde{O}(\frac{N^{3/2}\sqrt{d}}{\sqrt{T}})$. This yields the result. $\square$

A.3 Analysis of Algorithm 4

Recall Algorithm 4 is the modification of Algorithm 2 where we sample $\tilde{O}(\frac{N^2 \ln(NT)}{\gamma^2})$ times a point (instead of $O(\frac{\ln(T/(\alpha\gamma))}{\alpha^2\gamma^2})$) to build a γ-CI. In this section we present the proofs of Theorems 10 and 11. We only need to show that $\tilde{O}(\frac{N^2 \ln(NT)}{\gamma^2})$ samples are sufficient to build a γ-CI that holds with high probability. Afterwards it is easy to verify that the proofs of Theorems 6 and 7 go through.

Lemma 72. *To build a γ-CI for $\rho[F](x)$ that holds with probability at least $1 - \frac{1}{T^2}$ we need no more than $O(\frac{N \ln(N) \ln(\sqrt{N}T)}{\gamma^2})$ samples.*

Proof. Notice that

$$|\rho[X] - \hat{\rho}[X]| = |\sum_{n=1}^{N} \mu_n (C_{n/N}[X] - \hat{C}_{n/N}[X])| \leq \sum_{n=1}^{N} \mu_n |C_{n/N}[X] - \hat{C}_{n/N}|$$

Therefore, if we obtain γ-CI's for each term $|C_{n/N}[X] - \hat{C}_{n/N}|$ that hold with probability at least $1 - \frac{1}{NT^2}$ a union bound yields the result. From Theorem 1 we know that $O(\frac{N^2 \ln(\sqrt{N}T)}{n^2\gamma^2})$ samples suffice to build a γ-CI for $C_{n/N}[X]$ that holds probability at least $1 - \frac{1}{NT^2}$. Summing up the number of samples, approximating the sum with an integral and using a union bound yields the result. $\square$

We are now ready to prove the theorems.

Proof of Theorem 10. It is easy to see that the proof of Theorem 6 goes through if we set $h(\cdot) = \rho[F](\cdot)$ and we replace everywhere the number of times we sample a point $O(\frac{\ln(T/(\alpha\gamma))}{\alpha^2\gamma^2})$ with $\tilde{O}(\frac{N^2 \ln(T)}{\gamma^2})$. $\square$

Proof of Theorem 11. The proof follows from almost the same reasoning as in the proof of Theorem 7. We have

$$\begin{aligned}
&\rho[\{f_t(x_t)\}_{t=1}^T] - \min_{x\in X} \rho[\{f_t(x)\}_{t=1}^T] \\
&\leq \frac{N}{T}\sqrt{T}\sqrt{\frac{2}{\beta}\sum_{t=1}^T C_\alpha[F](x_t) - C_\alpha[F](x^*)} + CE \\
&\leq O(\frac{d^8 N^3}{\beta^{1/2}T^{1/4}}) + CE \quad \text{(with probability at least } 1 - \frac{1}{T})
\end{aligned}$$

where $CE = \rho[F](x^*) - \min_{x\in X} \rho[\{f_t(x)\}]$ and $x^* = argmin_{x\in X}\rho[F](x)$. Just as in the proof of Theorem 3 we can bound CE with probability at least $1 - \frac{2}{T}$ by $\tilde{O}(\frac{N^3/2\sqrt{d}}{\sqrt{T}})$. A union bound yields the result. □

A.4 Experimental Results

In this section we test the performance of Algorithms 1 and 2 and see if they behave as predicted. We first present experimental results for the 1-dimensional case and then for the general d-dimensional case.

A.4.1 The 1-Dimensional Case

We tested the algorithms against an instance generated the following way. We let $f(x,\xi) = \frac{1}{x} + (.05 - .04\xi)x^2$ where $\xi \sim U[0,1]$, that is ξ is sampled uniformly from the interval $[0,1]$, and $x \in X := [0.5, 6]$. By using Equation 5 some simple algebra yields $C_\alpha[F](x) = \frac{1}{x} + (.05 - .02\alpha)x^2$ with minimum occurring at $x^* = \frac{1}{(.1-.04\alpha)^{1/3}}$. Notice that with the closed form expressions it is easy to evaluate $\bar{\mathcal{R}}_T$. To evaluate $\mathcal{R}_T$ we will approximate the term $\min_{x\in X} C_\alpha[\{f_t(x^*)\}_{t=1}^T]$ with $C_\alpha[\{f_t(x^*)\}]$ which by Lemma 1 should not be too far. To compute empirical pseudo-regrets and regrets we observe the random losses and iterates generated by the algorithms when they are run for $T = 1,000,000$ rounds. Since the previous quantities may vary every time the algorithm is run, we will run each algorithm 25

times and average the outputs. All the parameters for Algorithm 1 where chosen optimally, the constants hidden by the O-notation were set to 1. The initial iterate was always set far from x^*, $x_0 = 5.8$ every time the algorithm was run.

In Figures A.1, A.2, A.3 and A.4 we compare the performance of Algorithm 1 versus Algorithm 2 at $\alpha = 1, .75, .25, .01$. It can be observed that, except for the case $\alpha = .75$, Algorithm 2 incurs less regret than its counterpart which is expected. To test whether the regrets and pseudo regrets are decaying at the predicted rates, we created log-log plots (i.e. take the logarithm of the values on both axes) and then measured the slope. Unfortunately, since Algorithm 2 is not "smooth" its log-log plots do not show the behavior we would like them to. For conciseness we omit the log-log plots of both algorithms and just report the slope of the log-log plots for Algorithm 1. From the reported values of the slopes of the log-log plots of the regret and pseudo regret curves, we can observe that the regret indeed decays at a slower rate that the pseudo regret as predicted by the theoretical results. The previous is more apparent for small values of α. We are unsure about why Algorithm 1 behaves better than expected when $\alpha = .75$.

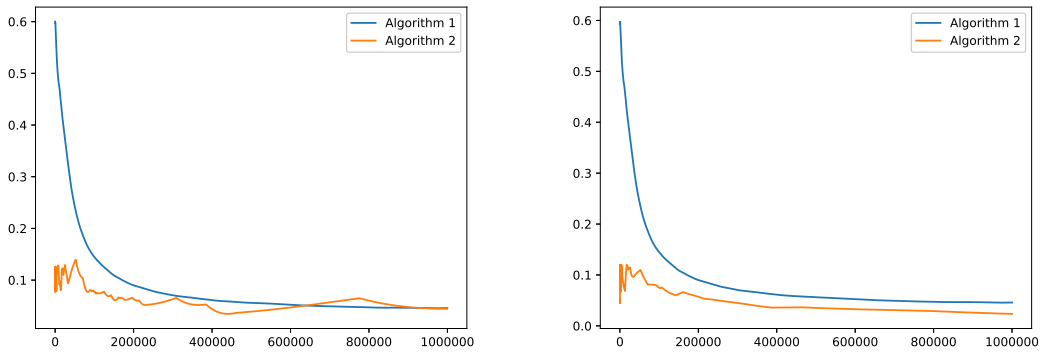

Figure A.1: Regret (left) and Pseudo Regret (right) of Algorithms 1 and 2 with $\alpha = 1$. The slopes of Algorithm 1 in the log-log plots are: -0.37 and -0.37 respectively.

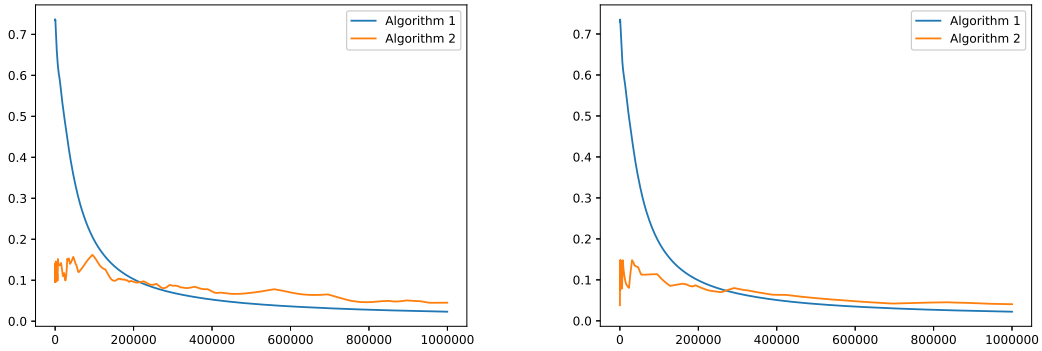

Figure A.2: Regret (left) and Pseudo Regret of Algorithms 1 and 2 with $\alpha = .75$. The slopes of Algorithm 1 in the log-log plots are: -0.49 and -0.51 respectively.

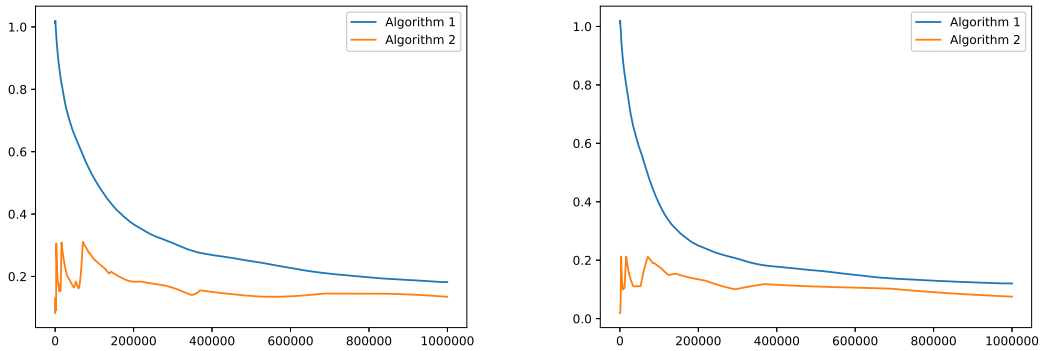

Figure A.3: Regret of Algorithms 1 and 2 with $\alpha = .25$. The slopes of Algorithm 1 in the log-log plots are: -0.24 and -0.31 respectively.

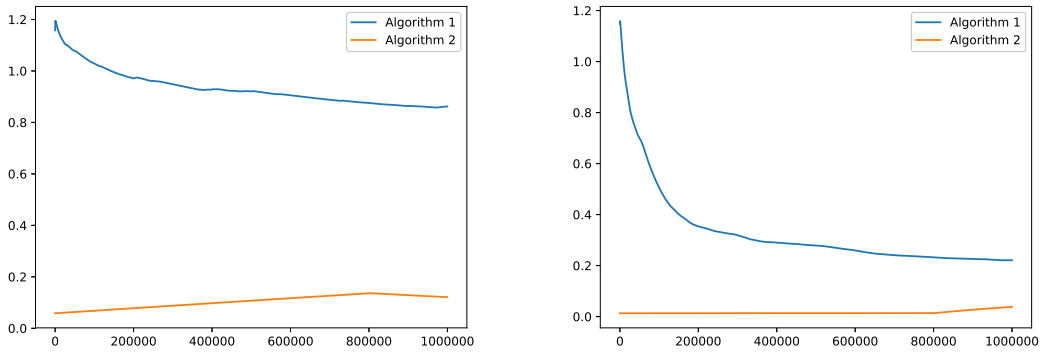

Figure A.4: Regret of Algorithms 1 and 2 with $\alpha = .01$. The slopes of Algorithm 1 in the log-log plots are: -0.04 and -0.24 respectively.

A.4.2 The d-Dimensional Case

Unfortunately Algorithm 2 does not work well in practice because of the large power on d that appears in the regret bound. Because of this we only present results for Algorithm 1. The instance for this section was generated as follows, let $f_i(x_i, \xi_i) = \frac{1}{x_i} + (.05 - .04\xi_i)x_i^2$, then define the loss $f(x, \xi) = \frac{1}{d}\sum_{i=1}^{d} f_i(x_i, \xi_i)$ with $\xi \sim U[0,1]^N$ (i.e. ξ is sampled uniformly from the d-dimensional [0,1] cube). It is easy to see that $C_\alpha[F](x) = \frac{1}{d}\sum_{n=1}^{d} \frac{1}{x_i} + (.05 - .02\alpha)x_i^2$ by the previous section. Even though the loss function is a summation of coordinate-wise independent functions the algorithm is not aware of it and thus can not exploit the structure. All the parameters of the algorithm were set optimally except for the constant hidden by the O-notation which was set to 1. The initial iterate was always set far from x^*, $x_0 = [5.8; ...; 5.8]$ every time the algorithm was run. In all the log-log plots we present below we compute the slope of each curve by using the second half of each curve.

When $\alpha = 1$, in Figures A.5 and A.6 we observe the regret and pseudo-regret of the algorithm "Gradient Descent without a Gradient", which uses Equation 2.1 to compute an estimate of the gradient and then performs a gradient step [53]. We notice that it behaves slightly better than Algorithm 1 (see Figures A.7 and A.8). (When $\alpha \neq 1$ we do not show how "Gradient Descent without a Gradient" performs because it is solving a different problem and thus the comparison does not make sense.) From Figures A.9, A.10, A.11, A.12, A.13, and A.14 we can observe several things. First, the dimension of the problem indeed affects the pseudo-regret and the regret negatively. Second, the smaller the α level the higher the pseudo-regret and regret. Third, the regret seems to vanish at a lower rate than the pseudo-regret (this can be seen by looking at the slopes of the log-log plots), this becomes clearer the smaller the α is, additionally in most plots the rate at which pseudo-regret and regret vanishes is not too far from the predicted ones. The previous may indicate that our analysis for the regret may be tight with respect to T. We notice that when $\alpha \neq 1$, at early rounds the regret seems to increase quickly before it starts dropping. The previous

occurs because at the beginning, even though we are in a region with bad $CVaR_\alpha[F]$, we have not observed many losses and thus the empirical $CVaR$ of the sequence of losses is not necessarily large. We are unsure about why at $\alpha = 0.75$ the regret seems to be vanishing a lot faster than predicted.

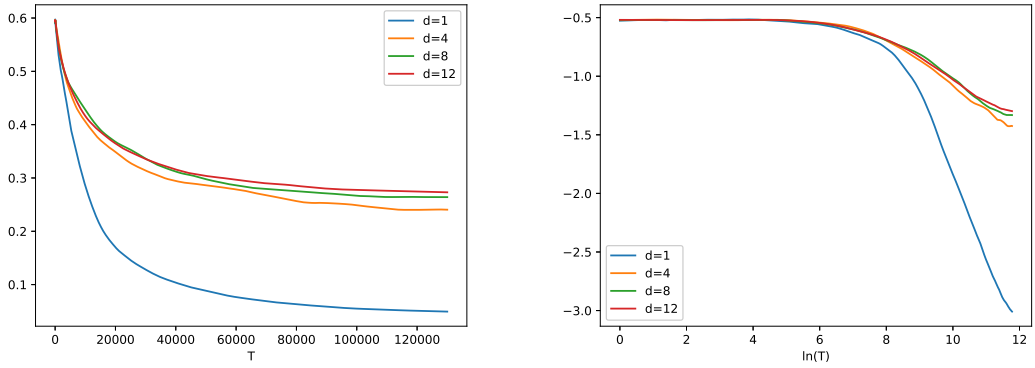

Figure A.5: Pseudo Regret (left) and its log-log plot (right) using Online Gradient Descent without a Gradient, $\alpha = 1.0$. The slopes of the curves in the log-log plots are: $\tilde{m}_1^1 = -0.42, \tilde{m}_4^1 = -0.15, \tilde{m}_8^1 = -0.13, \tilde{m}_{12}^1 = -0.13$

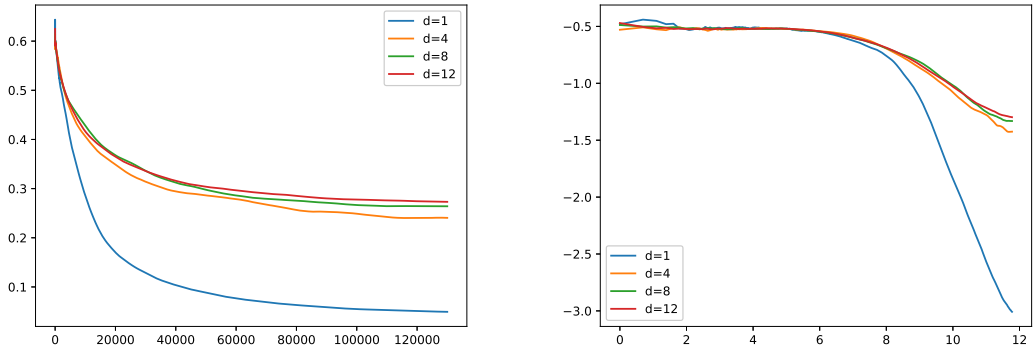

Figure A.6: Regret (left) and its log-log plot (right) using Online Gradient Descent without a Gradient, $\alpha = 1.0$. The slopes of the curves in the log-log plots are: $m_1^1 = -0.42, m_4^1 = -0.15, m_8^1 = -0.13, m_{12}^1 = -0.13$

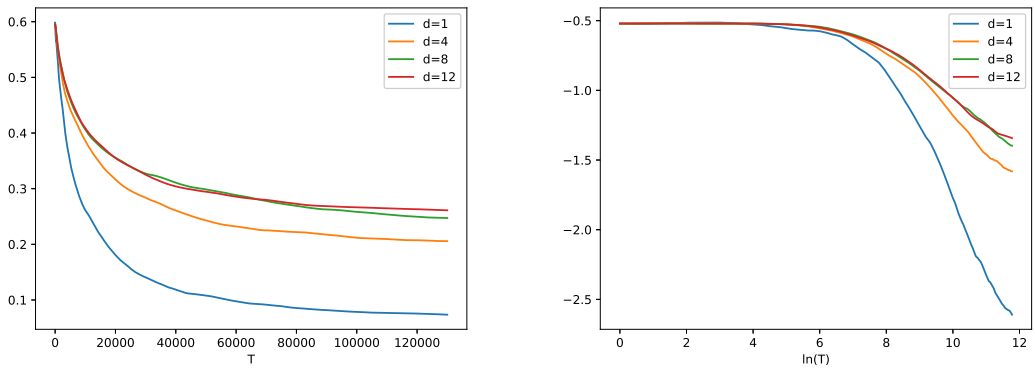

Figure A.7: Pseudo Regret (left) and its log-log plot (right) using Algorithm 1, $\alpha = 1$. The slopes of the curves in the log-log plots are: $\bar{m}_1^1 = -0.35, \bar{m}_4^1 = -0.17, \bar{m}_8^1 = -0.14, \bar{m}_{12}^1 = -0.13$

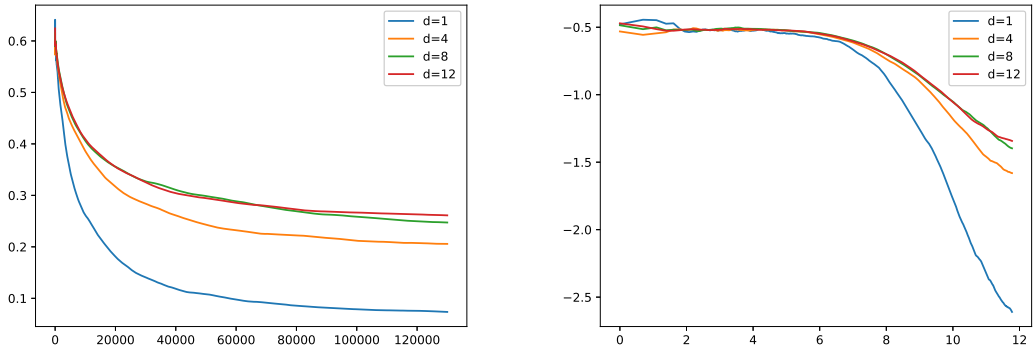

Figure A.8: Regret (left) and its log-log plot (right) using Algorithm 1, $\alpha = 1.0$. The slopes of the curves in the log-log plots are: $m_1^1 = -0.35, m_4^1 = -0.17, m_8^1 = -0.14, m_{12}^1 = -0.13$

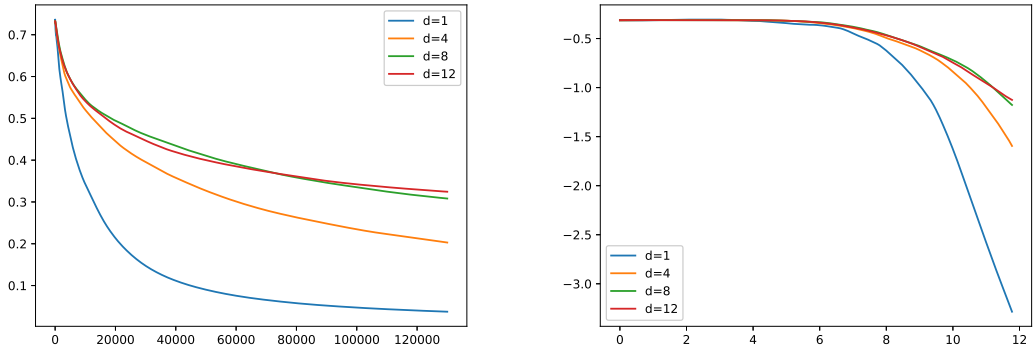

Figure A.9: Pseudo Regret (left) and its log-log plot (right) using Algorithm 1, $\alpha = 0.75$. The slopes of the curves in the log-log plots are: $\bar{m}_1^{75} = -0.50, \bar{m}_4^{75} = -0.21, \bar{m}_8^{75} = -0.14, \bar{m}_{12}^{75} = -0.13$

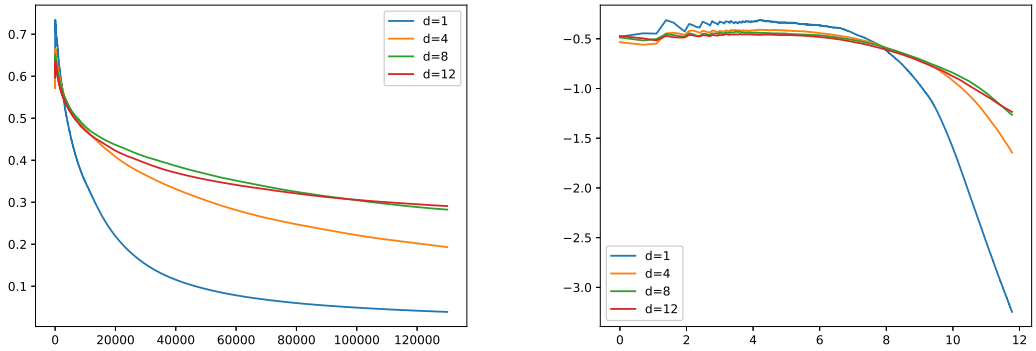

Figure A.10: Regret (left) and its log-log plot (right) using Algorithm 1, $\alpha = 0.75$. The slopes of the curves in the log-log plots are: $m_1^{75} = -0.49, m_4^{75} = -0.20, m_8^{75} = -0.13, m_{12}^{75} = -0.13$

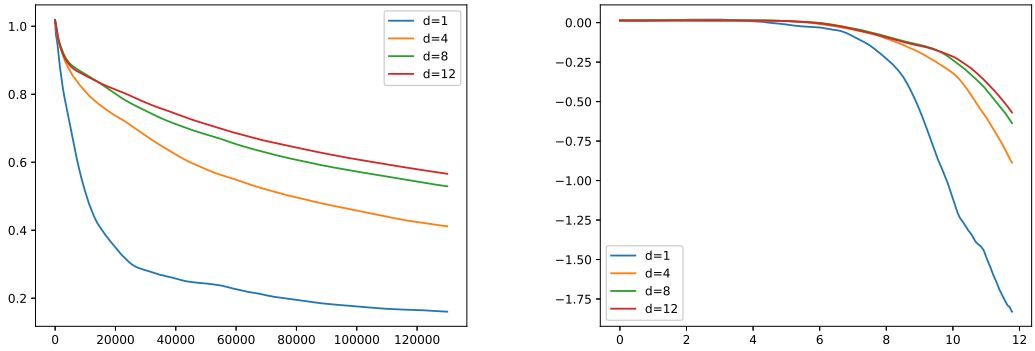

Figure A.11: Pseudo Regret (left) and its log-log plot (right) using Algorithm 1, $\alpha = 0.25$. The slopes of the curves in the log-log plots are: $\bar{m}_1^{.25} = -0.31, \bar{m}_4^{.25} = -0.15, \bar{m}_8^{.25} = -0.10, \bar{m}_{12}^{.25} = -0.09$

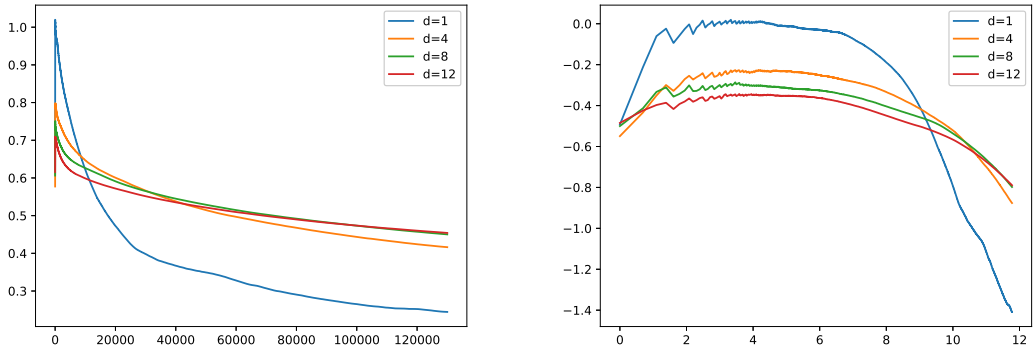

Figure A.12: Regret (left) and its log-log plot (right) using Algorithm 1, $\alpha = 0.25$. The slopes of the curves in the log-log plots are: $m_1^{.25} = -0.23, m_4^{.25} = -0.10, m_8^{.25} = -0.08, m_{12}^{.25} = -0.07$

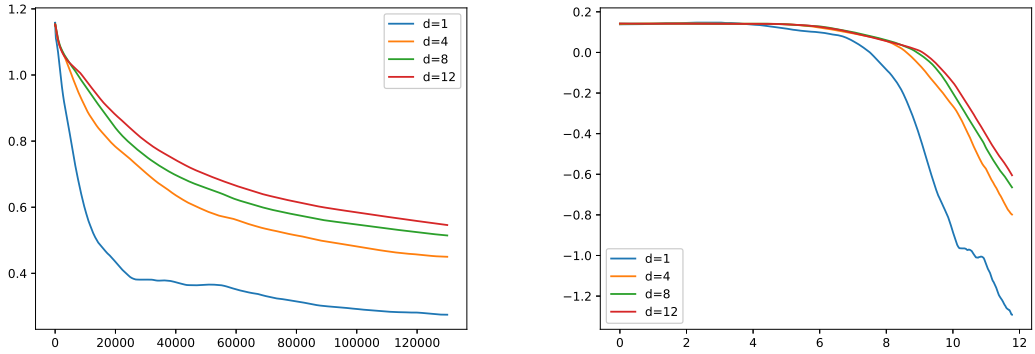

Figure A.13: Pseudo Regret (left) and its log-log plot (right) using Algorithm 1, $\alpha = 0.01$. The slopes of the curves in the log-log plots are: $\bar{m}_1^{.01} = -0.24, \bar{m}_4^{.01} = -0.15, \bar{m}_8^{.01} = -0.13, \bar{m}_{12}^{.01} = -0.12$

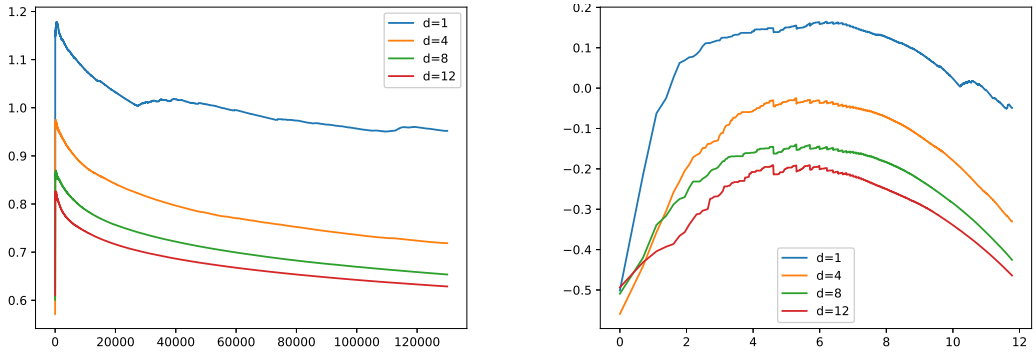

Figure A.14: Regret (left) and its log-log plot (right) using Algorithm 1, $\alpha = 0.01$. The slopes of the curves in the log-log plots are: $m_1^{.01} = -0.03, m_4^{.01} = -0.05, m_8^{.01} = -0.04, m_{12}^{.01} = -0.04$

A.5 Properties of the Pyramid Construction

The proofs of the next two lemmas can be found in [9]. Recall $\phi = arcos(c_2/d)$. We assume $d \geq 2$, because of this $\cos(\phi) = c_2/d$ and $\sin(\phi) = \sqrt{1 - c_2^2/d^2} \geq \cos(\phi)$. Also, recall that in epoch τ the initial simplex is contained in $\mathcal{B}(r_\tau)$ with $r_\tau = R_\tau/(c_1 d)$.

Lemma 73. *Let Π_k be the k-th pyramid constructed in any round of epoch τ.*

1. *The distance from the center of* $\mathcal{B}(r_\tau)$ *to the apex of* Π_k *is* $r_\tau \sin^{k-1}(\phi)$.

2. *The distance from the apex of* Π_k *to any vertex of the base of* Π_k *is* $r_\tau \sin^{k-1}(\phi)\cos(\phi)$.

3. *The height of* Π_k*, i.e. the distance from the apex to the base, is* $r_\tau \sin^{k-1}(\phi)\cos^2(\phi)$.

Lemma 74. *Let* Π *by any pyramid constructed in epoch* τ *with apex at distance* $r_\Pi \geq r_\tau/d$ *from the center of* $\mathcal{B}(r_\tau)$*. Let* $\mathcal{B}_\Pi$ *be the largest ball in* Π *centered at the center of mass* c *of* Π*.*

1. $\mathcal{B}_\Pi$ *has radius at least* $r_\Pi \cos^2(\phi)/(d+1) \geq r_\tau c_2^2/(2d^4)$.

2. *Let* $x \in \Pi$*, and let* $b \in \Pi$ *be the point on the face of* Π *such that* $c = \alpha x + (1-\alpha)b$ *for some* $0 < \alpha \leq 1$*. Then* $(1-\alpha)/\alpha \leq (d+1)d/c_2$.

APPENDIX B

DIFFERENTIALLY PRIVATE ONLINE SUBMODULAR OPTIMIZATION

B.1 Tree-Based Aggregation Protocol (TBAP)

The Tree-Based Aggregation Protocol is a tool for maintaining differentially private partial sums of vectors arriving in an online sequence. At each time t, TBAP outputs a noisy sum of the input vectors up to time t. This algorithm was first introduced by Chan et al. [38] and Dwork et al. [46], and adapted in its current form by Smith and Thakurta [121].

The algorithm works by maintaining a complete binary tree, where the d-dimensional input vectors are stored in the leaf nodes, and internal nodes in the tree store a noisy sum of all leaves in their sub-tree. At each time t, TBAP receives input z_t and updates the value of the t-th leaf node to be z_t. The algorithm also updates the value of each internal node affected by this change to be the updated sum plus noise drawn according to a high-dimensional analog of Laplace noise. The algorithm then outputs a noisy partial sum v_t of the nodes in the tree that approximately sum to z_t.

The sum at each internal node is $(\epsilon/\log_2 T)$-differentially private, and by construction each z_t affects only $\log_2 T$ nodes of the tree. By the *composition property* of differential privacy [45], the entire tree is ϵ-differentially private (Theorem 30).

Theorem 30 ([38, 46])**.** TBAP$(\{z_i\}_{i=1}^T, \mu, \epsilon)$ *is ϵ-differentially private for any $\mu > 0$ and any sequence of vectors $z_1, \ldots, z_T$ that each have L_2 norm at most μ.*

In addition to being private, TBAP also provides partial sums $v_t = \sum_{i=1}^t z_t$ that are accurate (with respect to the L_2 norm) up to additive $O(\frac{d\mu \log^2 T}{\epsilon})$. This is because the L_2 norm of the noise at each node is Gamma distributed with standard deviation $O(\frac{\sqrt{d}\mu \log T}{\epsilon})$, and each partial sum is computed using at most $\log T$ nodes in the tree.

Algorithm 17 Tree Based Aggregation Protocol: TBAP$(\{z_i\}_{i=1}^T, \mu, \epsilon)$

Input: Online sequence of vectors $z_1, ..., z_T \in \mathbb{R}^d$, μ : L_2-norm bound on each z_i, privacy parameter ϵ.
Output: Sequence of noisy partial sums $v_1, \ldots, v_n \in \mathbb{R}$

Initialize a binary tree A of size $2^{\lceil \log_2 T \rceil + 1} - 1$ with leaves $z_1, ..., z_T$
for t = 1, . . . , T **do**
  Accept z_t from the data stream.
  Let $P = \{z_t \to \cdots \to root\}$ be the path from z_t to the root.
  procedure Tree update
  Let Λ be the first node in P that is left-child in A. Let $P_\Lambda = \{z_t \to \cdots \to \Lambda\}$.
  for all nodes α in path P **do**
    $\alpha \leftarrow \alpha + z_t$
    if $\alpha \in P_\Lambda$ **then**
      $\alpha \leftarrow \alpha + \gamma$ where $\gamma \in \mathbb{R}^d$ is sampled by $\Pr[\gamma = \hat{\gamma}] \propto e^{-\frac{\|\hat{\gamma}\|_2 \epsilon}{\mu(\lceil \log_2 T \rceil + 1)}}$
    end if
  end for
  end procedure
  procedure Output private partial sum
  Initialize vector $v_t \in \mathbb{R}^d$ to zero. Let b be a $(\lceil \log_2 T \rceil + 1)$-bit binary representation of t.
  for $i = 1, \ldots, [\log_2 T + 1]$ **do**
    if bit $b_i = 1$ **then**
      if i-th node in P (denoted $P(i)$) is the left child in A, **then**
        $v \leftarrow v + P(i)$
      else
        $v_t \leftarrow v_t +$left sibling $P(i)$
      end if
    end if
  end for
  return noisy partial sum v_t
  end procedure
end for

APPENDIX C

LARGE SCALE MARKOV DECISION PROCESSES WITH CHANGING REWARDS

C.1 Bounding the problem dependent constant in Theorem 28

Consider the LP formulation of the l_1 projection problem of $\Phi\tilde{\theta}$ onto $\Delta^{\Phi}_{M,\delta}$.

$$\begin{aligned}
&\min_{\theta,u} \sum_{(s,a)} u(s,a) \\
&\text{s.t} \quad u(s,a) - [\Phi\theta](s,a) \geq -[\Phi\tilde{\theta}](s,a), \quad u(s,a) + [\Phi\theta](s,a) \geq [\Phi\tilde{\theta}](s,a), \\
&\qquad \mu = \Phi\theta, \quad \mu^\top 1 = 1, \quad \mu \geq \delta, \quad \mu^\top (P-B) \geq 0, \quad -\mu^\top (P-B) \geq 0 \\
&\qquad -\theta(i) \geq -W \quad , \quad \theta(i) \geq 0 \quad \forall i = 1, ..., d.
\end{aligned}$$

Fix any state action pair $(s', a') \in S \times A$ and change the constraint $\mu(s', a') \geq \delta$ for $\mu(s', a') \geq \delta + \gamma_{s',a'}$. Let $obj(\gamma_{s',a'})$ be the optimal value of the above LP with the constraint is replaced by $\mu(s', a') \geq \delta + \gamma_{s',a'}$. Let $\mu^*(\gamma_{s',a'})$ be the optimal solution to this problem. Let $\bar{\gamma}_{s',a'}$ be the maximum value of $\gamma_{s',a'}$ such that the LP above is feasible.

Some remarks are in order. First, for any $\gamma_{s',a'} \in [0, \bar{\gamma}_{s',a'}]$, it holds that $obj(\gamma_{s',a'}) \geq 0$. Second, $obj(\gamma_{s',a'})$ is a convex and increasing function in $\gamma_{s',a'}$. Third, a subgradient of $obj(\gamma_{s',a'})$ is given by the optimal dual variable associated with the constraint $\mu(s', a') \geq \delta + \gamma_{s',a'}$. Let us call this optimal dual variable $\lambda^*(\gamma_{s',a'})$. Since the above LP's objective is equivalent to $\|\mu - \Phi\tilde{\theta}\|_1$, using triangle inequality we have that $obj(\bar{\gamma}_{s',a'}) \leq \|\mu^*(\bar{\gamma}_{s',a'})\|_1 + \|\Phi\tilde{\theta}\| \leq 1 + \|\Phi\tilde{\theta}\| \leq 2$, where the last inequality holds since $(\Phi\tilde{\theta})^\top 1 = 1$.

We are ready to upper bound $\lambda^*(0)$ which is the subgradient of $obj(\gamma_{s',a'})$ for $\gamma_{s',a'} = 0$. Since $obj(\gamma)$ is an increasing function, we can upper bound $\lambda^*(0)$ with the slope of the line

that passes through the points $(0, obj(0))$ and $(\bar{\gamma}_{s',a'}, obj(\bar{\gamma}_{s',a'}))$. The slope of this line is $\frac{obj(\bar{\gamma}_{s',a'})-obj(0)}{\bar{\gamma}_{s',a'}}$. We have that

$$\lambda^*(0) \leq \frac{obj(\bar{\gamma}_{s',a'}) - obj(0)}{\bar{\gamma}_{s',a'}} \leq \frac{2 - obj(0)}{\bar{\gamma}_{s',a'}} \leq \frac{2}{\bar{\gamma}_{s',a'}},$$

where the last inequality holds since $\| \cdot \|_1 \geq 0$.

Let us now discuss in more detail the quantity $\bar{\gamma}_{s',a'}$. It turns out to be problem-dependent. For example, consider an MDP such that regardless of the action chosen by the player, it transitions to any state with equal probability and there is only one action at each state, then $\bar{\gamma}_{s',a'} = \frac{1}{|S|}$. Thus, the bound for $c_{S,A}$ becomes $c_{S,A} \leq 2|S|$, which depends linearly on $|S|$. Consider another example: suppose the MDP is such that for any state, there exists an action that allows us to remain in that state with probability 1 (a concrete case is the Markovian multi-armed bandit problem with the "retirement" option, see Whittle [133], Weber [132]). This implies that we can make the occupancy measure equal to a vector consisting of zeros of dimension $|S||A|$ with a 1 on any desired entry. Then, the analysis above shows that $\bar{\gamma}_{s',a'} = 1$.

BIBLIOGRAPHY

[1] Y. Abbasi, P. L. Bartlett, V. Kanade, Y. Seldin, and C. Szepesvári. Online learning in Markov decision processes with adversarially chosen transition probability distributions. In *Advances in Neural Information Processing Systems*, pages 2508–2516, 2013.

[2] Y. Abbasi-Yadkori, P. L. Bartlett, and A. Malek. Linear programming for large-scale Markov decision problems. In *International Conference on Machine Learning*, volume 32, pages 496–504. MIT Press, 2014.

[3] Y. Abbasi-Yadkori, P. L. Bartlett, X. Chen, and A. Malek. Large-scale Markov decision problems via the linear programming dual. *arXiv preprint arXiv:1901.01992*, 2019.

[4] J. Abernethy, K. A. Lai, K. Y. Levy, and J.-K. Wang. Faster rates for convex-concave games. *arXiv preprint arXiv:1805.06792*, 2018.

[5] J. D. Abernethy and J.-K. Wang. On Frank-Wolfe and equilibrium computation. In *Advances in Neural Information Processing Systems*, pages 6587–6596, 2017.

[6] J. D. Abernethy, E. Hazan, and A. Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. *In Proceedings of the 21st Annual Conference on Learning Theory (COLT)*, 2009.

[7] C. Acerbi and D. Tasche. On the coherence of expected shortfall. *Journal of Banking & Finance*, 26(7):1487–1503, 2002.

[8] I. Adler. The equivalence of linear programs and zero-sum games. *International Journal of Game Theory*, 42(1):165–177, 2013.

[9] A. Agarwal, D. P. Foster, D. J. Hsu, S. M. Kakade, and A. Rakhlin. Stochastic convex optimization with bandit feedback. In *Advances in Neural Information Processing Systems*, pages 1035–1043, 2011.

[10] N. Agarwal and K. Singh. The price of differential privacy for online learning. *arXiv preprint arXiv:1701.07953*, 2017.

[11] S. Agrawal, Z. Wang, and Y. Ye. A dynamic near-optimal algorithm for online linear programming. *Operations Research*, 62(4):876–890, 2014.

[12] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[13] K. J. Arrow, D. Blackwell, and M. A. Girshick. Bayes and minimax solutions of sequential decision problems. *Econometrica, Journal of the Econometric Society*, pages 213–244, 1949.

[14] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *focs*, page 322. IEEE, 1995.

[15] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multi-armed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.

[16] R. J. Aumann. Correlated equilibrium as an expression of bayesian rationality. *Econometrica: Journal of the Econometric Society*, pages 1–18, 1987.

[17] F. Bach. Learning with submodular functions: A convex optimization perspective. *Foundations and Trends in Machine Learning*, 6(2-3):145–373, 2013.

[18] A. Badanidiyuru, R. Kleinberg, and A. Slivkins. Bandits with knapsacks. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 207–216. IEEE, 2013.

[19] A. Badanidiyuru, B. Mirzasoleiman, A. Karbasi, and A. Krause. Streaming submodular maximization: Massive data summarization on the fly. In *Proceedings of the 20th ACM International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 671–680, 2014.

[20] D. Balduzzi, S. Racaniere, J. Martens, J. Foerster, K. Tuyls, and T. Graepel. The mechanics of n-player differentiable games. *arXiv preprint arXiv:1802.05642*, 2018.

[21] K. Ball. An elementary introduction to modern convex geometry. *Flavors of geometry*, 31:1–58, 1997.

[22] R. Bellman. A Markovian decision process. *Journal of Mathematics and Mechanics*, pages 679–684, 1957.

[23] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust optimization*, volume 28. Princeton University Press, 2009.

[24] A. Ben-Tal, E. Hazan, T. Koren, and S. Mannor. Oracle-based robust optimization via online learning. *Operations Research*, 63(3):628–638, 2015.

[25] D. Berthelot, T. Schumm, and L. Metz. Began: boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017.

[26] D. P. Bertsekas. *Dynamic programming and optimal control*, volume 2. Athena Scientific, Belmont, MA, 4 edition, 2012.

[27] D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-dynamic programming*. Athena Scientific, Belmont, MA, 1996.

[28] M. Bowling. Convergence and no-regret in multiagent learning. In *Advances in neural information processing systems*, pages 209–216, 2005.

[29] M. Bowling and M. Veloso. Convergence of gradient dynamics with a variable learning rate. In *ICML*, pages 27–34, 2001.

[30] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[31] A. Brock, T. Lim, J. M. Ritchie, and N. Weston. Neural photo editing with introspective adversarial networks. *arXiv preprint arXiv:1609.07093*, 2016.

[32] S. Bubeck, N. Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5 (1):1–122, 2012.

[33] S. Bubeck, R. Eldan, and Y. T. Lee. Kernel-based methods for bandit convex optimization. *arXiv preprint arXiv:1607.03084*, 2016.

[34] S. Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

[35] A. R. Cardoso and R. Cummings. Differentially private online submodular optimization. *arXiv preprint arXiv:1807.02290*, 2018.

[36] N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

[37] N. Cesa-Bianchi, Y. Mansour, and G. Stoltz. Improved second-order bounds for prediction with expert advice. *Machine Learning*, 66(2-3):321–352, 2007.

[38] T.-H. H. Chan, E. Shi, and D. Song. Private and continual release of statistics. *ACM Transactions on Information and System Security*, 14(3):1–24, 2011.

[39] K. Chatterjee. Markov decision processes with multiple long-run average objectives. In *International Conference on Foundations of Software Technology and Theoretical Computer Science*, pages 473–484. Springer, 2007.

[40] Y. Chen, L. Li, and M. Wang. Scalable bilinear pi learning using state and action features. *arXiv preprint arXiv:1804.10328*, 2018.

[41] V. Conitzer and T. Sandholm. Awesome: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. *Machine Learning*, 67(1-2):23–43, 2007.

[42] T. M. Cover. Universal portfolios. *Mathematical finance*, 1(1):1–29, 1991.

[43] D. P. De Farias and B. Van Roy. The linear programming approach to approximate dynamic programming. *Operations research*, 51(6):850–865, 2003.

[44] T. Dick, A. Gyorgy, and C. Szepesvari. Online learning in Markov decision processes with changing cost sequences. In *International Conference on Machine Learning*, pages 512–520, 2014.

[45] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the 3rd Conference on Theory of Cryptography*, TCC '06, pages 265–284, 2006.

[46] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum. Differential privacy under continual observation. In *Proceedings of the 42nd ACM Symposium on Theory of Computing*, STOC '10, pages 715–724, 2010.

[47] C. Dwork, A. Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[48] E. Even-Dar, S. M. Kakade, and Y. Mansour. Experts in a Markov decision process. In *Advances in Neural Information Processing Systems*, pages 401–408, 2005.

[49] E. Even-Dar, M. Kearns, and J. Wortman. Risk-sensitive online learning. In *ALT*, pages 199–213. Springer, 2006.

[50] E. Even-Dar, S. M. Kakade, and Y. Mansour. Online Markov decision processes. *Mathematics of Operations Research*, 34(3):726–736, 2009.

[51] K. Ferreira, D. Simchi-Levi, and H. Wang. Online network revenue management using thompson sampling. *Operations research*, 2018. forthcoming.

[52] C. Finn, I. Goodfellow, and S. Levine. Unsupervised learning for physical interaction through video prediction. In *Advances in neural information processing systems*, pages 64–72, 2016.

[53] A. D. Flaxman, A. T. Kalai, and H. B. McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 385–394. Society for Industrial and Applied Mathematics, 2005.

[54] B. for International Settlements. Basel iii: international regulatory framework for banks.

[55] D. P. Foster. Prediction in the worst case. *The Annals of Statistics*, pages 1084–1090, 1991.

[56] Y. Freund and R. E. Schapire. Game theory, on-line prediction and boosting. In *Proceedings of the ninth annual conference on Computational learning theory*, pages 325–332. ACM, 1996.

[57] Y. Freund and R. E. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1-2):79–103, 1999.

[58] S. Fujishige. *Direct Submodular Functions and Optimization*. Annals of Discrete Mathematics. Elsevier, 2005.

[59] P. Gajane, R. Ortner, and P. Auer. A sliding-window algorithm for Markov decision processes with arbitrarily changing rewards and transitions. *arXiv preprint arXiv:1805.10066*, 2018.

[60] N. Galichet, M. Sebag, and O. Teytaud. Exploration vs exploitation vs safety: Risk-aware multi-armed bandits. In *Asian Conference on Machine Learning*, pages 245–260, 2013.

[61] D. Goldfarb and M. J. Todd. Modifications and implementation of the ellipsoid algorithm for linear programming. *Mathematical Programming*, 23(1):1–19, 1982.

[62] I. Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

[63] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[64] O. Granichin. Stochastic approximation with input perturbation under dependent observation noises. - . *1*. . . , 4:27–31, 1989.

[65] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017.

[66] A. Gupta, K. Ligett, F. McSherry, A. Roth, and K. Talwar. Differentially private combinatorial optimization. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10, pages 1106–1125, 2010.

[67] J. Hannan. Approximation to bayes risk in repeated play. *Contributions to the Theory of Games*, 3:97–139, 1957.

[68] E. Hazan and S. Kale. Online submodular minimization. *Journal of Machine Learning Research*, 13:2903–2922, 2012.

[69] E. Hazan and S. Kale. An optimal algorithm for stochastic strongly-convex optimization. *Journal of Machine Learning Research*, 15:2489–2512, 2014.

[70] E. Hazan and Y. Li. An optimal algorithm for bandit convex optimization. *arXiv preprint arXiv:1603.04350*, 2016.

[71] E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2):169–192, 2007.

[72] E. Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

[73] D. P. Helmbold, R. E. Schapire, Y. Singer, and M. K. Warmuth. On-line portfolio selection using multiplicative updates. *Mathematical Finance*, 8(4):325–347, 1998.

[74] N. Ho-Nguyen and F. Kılınç-Karzan. The role of flexibility in structure-based acceleration for online convex optimization. Technical report, Carnegie Mellon University, 2016. Technical report, http://www. optimization-online. org/DB_HTML/2016/08/5571.html.

[75] R. A. Howard. *Dynamic programming and Markov processes*. John Wiley, 1960.

[76] N. Immorlica, K. A. Sankararaman, R. Schapire, and A. Slivkins. Adversarial bandits with knapsacks. *arXiv preprint arXiv:1811.11881*, 2018.

[77] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017.

[78] P. Jain, P. Kothari, and A. Thakurta. Differentially private online learning. In *Proceedings of the 25th Annual Conference on Learning Theory*, COLT '12, pages 1–34, 2012.

[79] S. Jegelka and J. Blimes. Submodularity beyond submodular energies: Coupling edges in graph cuts. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 1897–1904, 2011.

[80] A. Juditsky, A. Nemirovski, et al. First order methods for nonsmooth convex large-scale optimization, i: general purpose methods. *Optimization for Machine Learning*, pages 121–148, 2011.

[81] S. Junges, N. Jansen, C. Dehnert, U. Topcu, and J.-P. Katoen. Safety-constrained reinforcement learning for mdps. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 130–146. Springer, 2016.

[82] A. Kalai and S. Vempala. Efficient algorithms for universal portfolios. *Journal of Machine Learning Research*, 3:423–440, 2002.

[83] A. Kalai and S. Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

[84] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[85] N. Kodali, J. Abernethy, J. Hays, and Z. Kira. On convergence and stability of gans. *arXiv preprint arXiv:1705.07215*, 2017.

[86] E. Koutsoupias. The k-server problem. *Computer Science Review*, 3(2):105–118, 2009.

[87] A. Krause and C. Guestrin. Submodularity and its applications in optimized information gathering. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(4):1–20, 2011.

[88] J. Křetínskỳ, G. A. Pérez, and J.-F. Raskin. Learning-based mean-payoff optimization in an unknown mdp under omega-regular constraints. *arXiv preprint arXiv:1804.08924*, 2018.

[89] S. Kusuoka. On law invariant coherent risk measures. In *Advances in mathematical economics*, pages 83–95. Springer, 2001.

[90] T. L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

[91] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, volume 2, page 4, 2017.

[92] N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine learning*, 2(4):285–318, 1988.

[93] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.

[94] M. L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine Learning Proceedings 1994*, pages 157–163. Elsevier, 1994.

[95] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet. Are gans created equal? a large-scale study. *arXiv preprint arXiv:1711.10337*, 2017.

[96] Y. Ma, H. Zhang, and M. Sugiyama. Online Markov decision processes with policy iteration. *arXiv preprint arXiv:1510.04454*, 2015.

[97] O.-A. Maillard. Robust risk-averse stochastic multi-armed bandits. In *International Conference on Algorithmic Learning Theory*, pages 218–233. Springer, 2013.

[98] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley. Least squares generative adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2813–2821. IEEE, 2017.

[99] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*, 2016.

[100] M. Mitrovic, M. Bun, A. Krause, and A. Karbasi. Differentially private submodular maximization: Data summarization in disguise. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 2478–2487, 2017.

[101] O. Morgenstern and J. Von Neumann. *Theory of games and economic behavior*. Princeton university press, 1953.

[102] J. Nash. Non-cooperative games. *Annals of mathematics*, pages 286–295, 1951.

[103] A. Nemirovskii, D. B. Yudin, and E. R. Dawson. Problem complexity and method efficiency in optimization. 1983.

[104] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

[105] G. Neu, A. György, C. Szepesvári, and A. Antos. Online markov decision processes under bandit feedback. *IEEE Transactions on Automatic Control*, 59(3):676–691, 2014.

[106] G. Neu, A. Jonsson, and V. Gómez. A unified view of entropy-regularized Markov decision processes. *arXiv preprint arXiv:1705.07798*, 2017.

[107] N. Noyan and G. Rudolf. Kusuoka representations of coherent risk measures in general probability spaces. *Annals of Operations Research*, 229(1):591–605, 2015.

[108] A. Pichler and A. Shapiro. Uniqueness of kusuoka representations. *arXiv preprint arXiv:1210.7257*, 2012.

[109] M. L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[110] S. J. Reddi, S. Kale, and S. Kumar. On the convergence of adam and beyond. 2018.

[111] H. ROBBINS. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 55:527–535, 1952.

[112] R. T. Rockafellar and S. Uryasev. Optimization of conditional value-at-risk. In *Journal of Risk*. Citeseer, 2000.

[113] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

[114] A. Sani, A. Lazaric, and R. Munos. Risk-aversion in multi-armed bandits. In *Advances in Neural Information Processing Systems*, pages 3275–3283, 2012.

[115] A. Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, 1998.

[116] S. Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

[117] S. Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.

[118] S. Shalev-Shwartz, O. Shamir, N. Srebro, and K. Sridharan. Stochastic convex optimization.

[119] A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on stochastic programming: modeling and theory*. SIAM, 2009.

[120] S. Singh, M. Kearns, and Y. Mansour. Nash convergence of gradient dynamics in general-sum games. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 541–548. Morgan Kaufmann Publishers Inc., 2000.

[121] A. Smith and A. Thakurta. (Nearly) optimal algorithms for private online learning in full-information and bandit settings. In *Proceedings of the 26th International*

Conference on Neural Information Processing Systems, NIPS '13, pages 2733–2741, 2013.

[122] J. C. Spall. A one-measurement form of simultaneous perturbation stochastic approximation. *Automatica*, 33(1):109–112, 1997.

[123] J. T. Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks. *arXiv preprint arXiv:1511.06390*, 2015.

[124] E. Takimoto and M. K. Warmuth. Path kernels and multiplicative updates. *Journal of Machine Learning Research*, 4:773–818, 2003.

[125] W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

[126] D. M. Topkis. *Supermodularity and complementarity*. Princeton University Press, 2011.

[127] S. Vakili, K. Liu, and Q. Zhao. Deterministic sequencing of exploration and exploitation for multi-armed bandit problems. *IEEE Journal of Selected Topics in Signal Processing*, 7(5):759–767, 2013.

[128] J. Von Neumann. Die zerlegung eines intervalles in abzählbar viele kongruente teilmengen. *Fundamenta Mathematicae*, 1(11):230–238, 1928.

[129] V. G. Vovk. Aggregating strategies. *Proc. of Computational Learning Theory, 1990*, 1990.

[130] A. Wald. Sequential analysis, j. wiley & sons, new york. 1947.

[131] M. Wang. Primal-dual pi learning: Sample complexity and sublinear run time for ergodic Markov decision problems. *arXiv preprint arXiv:1710.06100*, 2017.

[132] R. Weber et al. On the Gittins index for multiarmed bandits. *The Annals of Applied Probability*, 2(4):1024–1033, 1992.

[133] P. Whittle. Multi-armed bandits and the Gittins index. *Journal of the Royal Statistical Society: Series B (Methodological)*, 42(2):143–149, 1980.

[134] J. Y. Yu and S. Mannor. Online learning in Markov decision processes with arbitrarily changing rewards and transitions. In *2009 International Conference on Game Theory for Networks*, pages 314–322. IEEE, 2009.

[135] J. Y. Yu and E. Nikolova. Sample complexity of risk-averse bandit-arm selection. In *IJCAI*, pages 2576–2582, 2013.

[136] J. Y. Yu, S. Mannor, and N. Shimkin. Markov decision processes with arbitrary reward processes. *Mathematics of Operations Research*, 34(3):737–757, 2009.

[137] B. Zhang, N. Wang, and H. Jin. Privacy concerns in online recommender systems: Influences of control and user data input. In *Proceedings of 10th Symposium On Usable Privacy and Security*, SOUPS '14, pages 159–173, 2014.

[138] A. Zimin and G. Neu. Online learning in episodic Markovian decision processes by relative entropy policy search. In *Advances in Neural Information Processing Systems*, pages 1583–1591, 2013.

[139] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 928–936, 2003.